Aladdin 4D

Version 5.0

Software:	**Thomas Krehbiel** **Greg Gorby**
Manual:	**Jay Gross** **Greg Gorby**
Illustrations:	**Harv Laser**
Layout:	**Kermit Woodall**
Direction:	**Bob Fisher**

Dedicated to Beverly and Greg Gorby

Table of Contents

Aladdin 4D Technical and Customer Support

Registered users may get installation assistance and report problems by calling Nova Designs' Aladdin 4D Support Line at 804-282-6528 during the hours of 1pm-5pm (Eastern Time), weekdays excluding holidays. Please respect the support line's hours. Calls will not be taken during any other hours. Before calling, be sure you have your serial number (found on the first install disk), and if possible have Aladdin 4D running on your Amiga near the phone when you call. Please read the appropriate sections of the manual before calling, as most support questions can be answered simply by referring to the manual.

If you wish to fax, the number is 804-282-3768, 24 hours. If you have Internet access, you might wish to subscribe to the Aladdin 4D mailing list. In addition, you can email Aladdin 4D Tech Support directly, or by clicking the appropriate button on Nova Design's site on the World Wide Web:

The Nova Design Home Page URL is: **http://www.novadesign.com**
Email address is: **kermit@novadesign.com**

The mailing address is:

Nova Design, Inc.
1910 Byrd Avenue, Suite 204
Richmond, VA 23230
USA

Telephones:

Dealer Sales	804-282-5868
Problem Reporting	804-282-6528
Fax	804-282-3768
Order Line	800-IMAGE-69 or 804-282-1157

User Support and Tutorials

We are looking for user-contributed articles and tutorials on the use of Aladdin 4D. Articles can be written towards first time users or advanced users. These articles will form the basis of a special tutorial area on the Aladdin support pages on the Internet and may be published in printed form as well!

If you are interested in contributing your article or tutorial, please contact us at our mailing address, support lines, or email us.

Package Contents

When you open the box, you should find:

Manual
Registration Card
One set of Installation Disks

Important!

Aladdin 4D is written specifically for Amigas equipped with a math coprocessor. If you do not have a math coprocessor it will not run, period.

Math coprocessors are included in most incarnations of the 68040 microprocessors - but not all of them - and on all Amiga 3000's, and in most 68030- and 68020-based accelerators - but not all of them. Aladdin 4D won't work with 68EC020, 68EC30, or 68EC40 microprocessors, because these chips do not have an on-board math "floating point unit" - or FPU. Some Amiga accelerators add an FPU, optionally. Aladdin 4D is internally optimized for the math limitations of 68040 processors.

The program also requires an Amiga powered by a 68020 microprocessor or higher. A regular 68000, like an unaccelerated Amiga 500, won't do. To get fast renders, you'll need faster, better processors like the 68040. Aladdin 4D is designed to be used with a hard disk drive.

Product Registration

Be sure to complete and mail the enclosed registration card. With the information on the card, we will be able to notify you of future upgrades, keep you posted on bug fixes, and let you know of other products. If you do not register, you will NOT have access to the technical support line.

License Agreement/Warranty

Read carefully the terms and conditions in ~~Appendix A~~**the Software License** prior to using these programs. Any use of any of these programs whatsoever indicates your acceptance of these terms and conditions.

Install

Aladdin 4D is designed to be used from a hard disk drive. You need enough empty space to hold the program and its support files, plus any pictures you need to store.

Do not use the original disks other than to install the program. You should immediately make one (1) copy of each of the original disks. Store the original disks in a safe place. Aladdin 4D installation disks are not copy protected. This is for your convenience. You may not lend, give, rent, sell or otherwise distribute these programs. Such action is illegal, against federal copyright laws and punishable.

If you have not done so, take the time now to read the terms and conditions in tAppendix he Software LicenseA. You are legally bound by these terms and conditions and we will actively pursue any violations.

The install disks

Instructions for installing Aladdin 4D are contained in a "readme" file on the disk that is labeled "Aladdin_Disk1." Put the disk in any Amiga floppy drive and double click the icon for the Readme file. Then, simply follow the instructions in the readme.

Memory and Hardware requirements

Computer: Aladdin 4D will run on any properly configured Amiga that has sufficient memory ("ram"), a hard disk drive, a math coprocessor, and a 68020 microprocessor or higher. To do the kind of intensive processing that 3D rendering and animation entails, you'll be much happier with a 68040 or better Amiga. The 68040 and above microprocessors have math capabilities built into the chip, and will work best.

Operating system: Aladdin 4D requires AmigaDOS Version 2.1 or above.

Memory: Aladdin 4D requires a minimum of only 2 megabytes of memory.

Drives: At least one 3.5" Amiga floppy disk drive is required.

Recommended: Aladdin 4D is a professional, calculation intensive 3D rendering and animation program. Although not required for operation, we recommend at least a 68040 processor with math coprocessor, 10 megs of memory and substantial hard drive capacity.

Virtual Memory

Aladdin 4D wants "real" memory, not "virtual" memory as performance will degrade considerably if you're swapping memory to and from a hard drive during renders.

Introduction

Welcome to Aladdin 4D, a program for making pictures out of math. You give the program a group of objects arranged the way you like in a three-dimensional space, and it calculates major amounts of 3D geometry till it comes up with the "view" of your objects you would see from where you sit at the Amiga. That's the basic plan in a nutshell - there's much more. For example, because the objects are rooted in math, Aladdin 4D can also create animations by calculating multiple frames, recalculating the objects' positions between "key frames" you establish, and along paths you define.
How Aladdin 4D does all this, and how it you relate to it, in order to tell if what you want it to do, are the subjects of this manual.

Using Aladdin 4D isn't difficult. The program offers as much user control as possible over the way the final pictures will look, so there are lots of options, adjustments, and parameters, resulting in many, many choices at every turn in the settings menus. After some experience and reading, however, you'll soon learn what the parameters do - they've been given English names on purpose - and it won't look so complicated as it does at first.

Extern Handlers

Extern handler hooks permit external programs to alter the database. This is the same as for external tools, but the extern handlers are able to do far more. The main difference is that the extern handlers can alter the database between frames in an animation! This presents many new possibilities in rendering.

Aladdin 4D comes with some demonstration Extern Handlers. Refer to the Reference sections for particulars.

ARexx

Many of Aladdin 4D's functions are controllable by ARexx from outside the program. Full documentation of the ARexx functionality is in the program's help system. Just hit the <Help> key, and look for ARexx in the online documentation.

Aladdin 4D Icon Tooltypes

The icons for the Aladdin 4D program can contain information about the way you want the program to behave when it starts up. You can change these from their defaults any time, and of course you can change many of the things they control from within the program. To change an icon's tooltype on the Amiga, click the icon once with the left mouse button - it will light up - then pull down the menu item "Information" from the Workbench. The window that opens gives you information about the program, and offers
a window labeled "tooltypes" in which you can scroll the available options. Amiga tooltypes are CASE SPECIFIC. They must be entered EXACTLY as listed here, paying attention to whether the words are caps or lowercase. If entering a Tooltype appears to have no effect, check to see whether you've typed it correctly, without extra spaces or mistaken capitalization.

Available tooltypes are already listed in this box for you. To enable one, highlight it for editing, remove the parentheses around it and supply any parameters it needs. Hit <Return>, and when you save the

icon's information window, the new tooltype will be active next time you run Aladdin 4D. You don't have to reboot your Amiga to get them to take effect, but they don't apply until you next run the program.

Aladdin 4D recognizes the following tooltypes. Don't worry if you don't understand what they do now - they'll all be clear when you've learned to use the program.

FONTNAME=<name>.font
— Font name to use for new look windows

FONTSIZE=<size>
— Font size to use for new look windows

MENUFONTNAME=<name>.font
— Font name to use for menus

MENUFONTSIZE=<size>
— Font size to use for menus

SCREENINTERLEAVED
— Under AmigaDOS 3.0, this opens an interleaved screen which greatly reduces bitplane flicker during screen updates. This has no effect for CyberGraphX screens, however.

SCREENDEPTH=<depth>
— Number of bitplanes to use for Editor screen (3 to 8). The default is 3. Additional bitplanes currently are not used, although if you use a CyberGraphX screen for the Editor, eight bitplanes may be faster than anything else.

FASTEDITOR
— Use one bitplane screen mode for the Editor window. This results in a very fast, but bland display. However, this also eliminates bitplane flicker.

NOSORTPOLYGONS
— Setting this tooltype bypasses the lengthy polygon sorting step that occurs before rendering begins. If you see problems occurring in extremely complex renderings, turn this back off (that is, delete it or put brackets around it in the Tooltypes list).

ROTATION
— Enable the animated, rotating view on startup. By default it is disabled.

TIPFONTNAME=<name>.font
— Font name to use for tool tips

TIPFONTSIZE=<size>
— Font size to use for tool tips

GROUNDGRID
— Controls whether the ground grid is on or off when the program opens. You can turn this on or off within the program. This tooltype simply lets you have it put there (or not) when you start Aladdin 4D.

Default Path Tooltypes

You can set up tooltypes in Aladdin 4D's icon to establish default places (paths) where the program will look for drawings, bitmap textures, CSplines, convolves and 4Dfonts. Texture and Attribute Lists are saved from the loaders themselves, and will always go back to the last directory you loaded from or saved to (even after shutting off the computer).

These default directories are initialized in the file requester as appropriate to the type of file you're loading or saving. For instance, the default drawing directory is "Aladdin 4D:drawings/", and the default texture load (bitmap) directory is "Aladdin 4D:textures/". You can change these to any device/directory you want by editing the tooltypes to match your system.

The logical device Aladdin 4D: is always present in the file requester's "device" list. This makes it easy to get to the program's "home" directory, no matter where you've placed it. The Aladdin 4D logical device is created during the installation process. You can change where it points by editing your S:User-Startup file. It's easy to spot - just look for the "BEGIN Aladdin 4D" flag.

ALL of the possibilities are in the Tooltypes list for Aladdin 4D as supplied on the distribution disks. The ones which appear without parentheses around them are in effect by default. If you wish to enable one, click it with the left mouse button, edit out the parentheses in the string requester below, and hit <Return>. To disable one, put parentheses around it. When you're done with changes to the Tooltypes, click "Save" to record your changes to the icon itself. Any changes you've made will be in effect next time you run Aladdin 4D by double clicking the icon.

The Application Icon

One more word about icons... While Aladdin 4D is running, it puts an "Appicon" on your Workbench screen, and installs itself in your Amiga's Workbench Tools pull-down menu. The icon gives you drag-and-drop functionality from the Workbench, without troubling with the program's Load requester. Simply drag an Aladdin 4D drawing (or one of its compatible loadable format files) over the Appicon, and let go of the mouse button. The program will load the item, just as though you'd called up its file requester and dug through the directories to find you file.

In addition, you can bring Aladdin 4D to the front with either the Tools menu item or the icon. You cannot run more than one incarnation of the program at one time - the system will tell you if the program is already running.

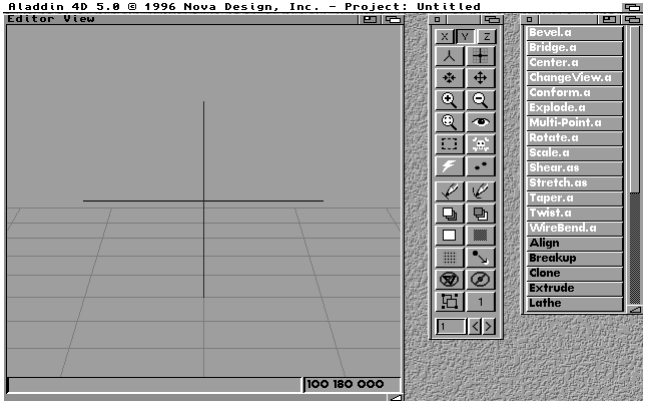
QUICKSTART

Aladdin 4D is a landmark program for the Amiga. Because of its complexity and the uniqueness of many of its tools and approaches, you might find yourself spending considerable time learning it. To help introduce you to the program's capabilities, we have included example drawings for you to load and render.

The first drawing is not very demanding on your system, but the second one "shows off" a little bit. These drawings will give you a small idea of what Aladdin 4D can do - and what you can make it do, when you gain experience to put your own ideas into phosphor.

In working through this manual, you should already be familiar with operating your Amiga - such as running programs, selecting gadgets and menu items, handling the mouse and keyboard, and selecting paths and filenames to load and save files. If you haven't yet acquired these skills, please make your life simpler by consulting your computer's manual for some basic training before you start trying to learn Aladdin 4D.

Getting started with Aladdin 4D

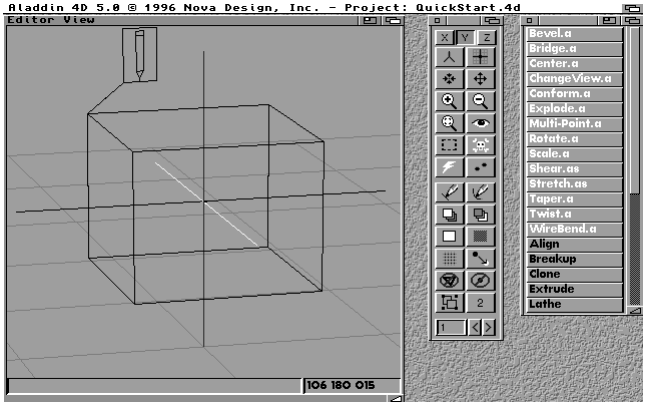


Start the program by double clicking the Aladdin 4D icon. Aladdin 4D opens on an Amiga custom screen within which it opens windows of its own which you can move around, resize, and arrange to suit yourself. The largest is the Edit Window for the project you're working on - and the smaller windows contain palettes of tools and options that you use to perform various operations on a project's wireframe objects. These include tools for creating objects, and cutting, conforming, and shaping them into the look you want, as well as applying various kinds of textures, illuminating them with lights, generating backgrounds, and so forth. In addition, there are many pull-down menus, available by pointing to the top bar of the program's window and holding down the right mouse button.

The Editor window can be full or partial screen, and you can use the "iconify" gadgets (if present) in the screens' title bars to shrink or enlarge them at will. The windows' front/back gadgets let you stack them up out of the way as you work. The tool palettes are movable and resizeable, for your convenience, too. If you shrink a window smaller than the information it contains, a scroll gadget will appear on its right-hand side, so you can scroll the contents of the window to reach the extra items in it.

You can set up the editor window to your liking with the Editor Settings requester, accessible from the Settings pull-down menu, but we'll fine-tune it later on.

Loading a drawing



When Aladdin 4D is on screen, from its leftmost pull-down menu "Project", choose "Open" and from the examples that are provided with the program choose "Quickstart.4D." As a project loads, the program puts up a little informational window which lists the possible components of a drawing - bitmaps, etc. - and displays a progress indicator to let you know what's being loaded and how long it's expected to take to finish. The Quickstart project is a simple one, so after only a few seconds of drive activity - and several progress meters - you'll see a wireframe view of... well, it's just a cube. The cube will appear in the "Editor View" window.

Is the cube drawing loaded? Well, so far, this is very unimpressive. How could we hope to get you excited with a simple cube on the screen? As you'll see shortly, this cube has hidden beauty. Its appeal is in the textures that are applied to it, and what it does with them during an animated sequence. Before looking into the rendering options, however, let's set the display mode - don't touch anything else just yet.

If you have one of the supported 24-bit display cards installed, or if you have an Amiga AGA machine, or a DCTV display device, do these next steps.

In order for Aladdin 4D to work with any of these display devices (except AGA), you must have told the program to support them during the install process. If you didn't do that, they won't appear among the options for screen modes to which you can render. To get them there, you must re-install the program and click the appropriate options for the hardware you have. Of course, you must also have the hardware itself properly installed in your Amiga. If you don't have any of these, or if you don't want to use them just now, skip this paragraph. Your Amiga's display mode is already set up for a standard Amiga display.

Aladdin 4D maintains multiple display mode settings all the time. One is the mode you're looking at when the program's user interface is on the screen, and another is the screen mode (or display device's mode) that the program will render its pictures to. You can change these any time you like, but in the interest of speeding up your workflow, it's a great help to have easy control over the render screen's parameters, so you can do quick low-resolution testing.

It is acceptable to have a larger screen size than is displayable by your monitor type. You can autoscroll the screen using the mouse; just move the pointer to the edge of the screen, and the Amiga will take care of the scrolling for you. Larger screens consume memory, however, so if you're short on that resource, it's better to stay conservative with screen sizes.

Aladdin 4D's configurable display feature can be used to make letterbox animations as well as for quick testing. When rendering to OpalVision, or similar boards, the image size is the operative size. The screen size is not used in the same way as in normal Amiga screen modes.
It is also perfectly acceptable to use different screen sizes and settings for Editor, Preview, and Render. When reloading saved drawings, the image sizes are loaded from the drawing. The Editor's screen mode is not loaded, but Preview and Render screen modes are.

Within Aladdin 4D, double-clicking (or single right mouse button clicking) most tool items will bring up the appropriate requester~~.~~ Within requesters brought up by the pull-down menus, a single click is usually sufficient~~.~~ Aladdin 4D offers multiple ways to accomplish the settings you want to make~~.~~ That is, you can click the tools or pull down the menus as most convenient to you, to make the settings changes you want.

Render Screen Mode

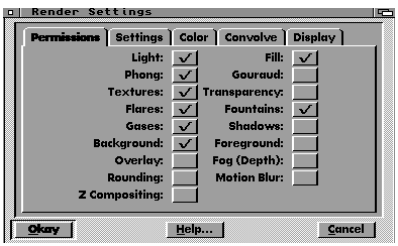
To change the display mode in which Aladdin 4D renders, pull down the Render Screen item from the "Render" menu in the menu bar (right mouse button)~~.~~ This will produce the usual Amiga screen mode requester, in which you can select display type, depth, etc. - or the appropriate mode to your add-on display hardware.

IMPORTANT!

You must have the mouse pointer in the menu bar area of the screen to get to the pull-down menus~~.~~ Aladdin 4D uses the right mouse button for other purposes when the mouse isn't pointing to the menu bar area~~.~~ If the menu bar says only "Aladdin 4D 5.0," click the Editor window to activate the menus.

The title bar will list the name of your active project when its menus are functional - in this case, "Quickstart.4D"~~.~~ If one of the other windows is active, it will display only "Aladdin 4D 5.0."

Render Settings - quick tour



Just below Render Screen in the Render menu is "Render Settings." ~~This are~~This is the nitty-gritty of the rendering specification for Aladdin 4D~~.~~ It's all much too huge to go into right now, but let's take a quick peek in there, anyway, to get the feel of what's going to happen.

Selecting Render Settings from the Render menu brings up a requester that lists the many available options for rendering~~.~~ The Render Settings requester is tabbed into groups of parameter settings, so lots of options can be accessed quickly and simply~~.~~ Click one of the tabs (Permissions, Settings, Color, Convolve, Display) to edit the settings~~.~~ We'll visit the other tab items in turn, but for now, let's just duck into the "Display" tab for a quick look around.

When you click the Display tab, Aladdin 4D provides a requester containing settings for the Display mode and such~~.~~ Select a screen mode with the top ~~rotary gadget~~cycle gadget~~.~~ On the Amiga, gadgets may rotate among two or more choices for a particular setting~~.~~ You choose by clicking the gadget one or more times to move through the available options~~.~~ This gadget has the following options, displayed in turn as you click it:

Amiga Screen - a picture displayable on your Amiga
DCTV - third party external hardware that uses a special Amiga screen mode to do its thing.
Opalvision - third-party 24-bit display hardware
Resolver - third-party display hardware
Retina - third-party display hardware
VideoToaster - NewTek's VideoToaster/Flyer display hardware
None (Render to Disk) - no display needed, just make a file on disk.

For any of the devices which need it, a "mode" for the display should be set up with the next gadget. This "mode" is a modifier to the first choice. It indicates whether a screen is to be high-resolution, HAM (in the case of Amiga displays), interlaced, etc. The available modes are listed in a separate requester when you click this gadget. The mode's pixel width and heights are displayed in the box next to the gadget, and you can also simply edit them, if you like. The pixel counts will change based on the "Overscan" setting, too.

A ~~rotary gadget~~cycle gadget cycles through the available options for overscan. You can also simply type arbitrary numbers into the text boxes under the overscan gadget - but the device you're rendering to must support the resolution you choose. A horizontal slider chooses the number of colors, and limits its range to the maximum number of colors supported by the "mode" you've chosen. If you don't see enough colors here, choose another mode or another hardware display card.

If you select "None" for Aladdin 4D's render screen mode, you can then type in custom settings for pixel width and height.

Watch out for the limitations and requirements of your intended display hardware if you decide to edit these pixel counts. Also Aladdin 4D likes screens whose dimensions are evenly divisible by 16.

If you select DCTV, be sure to click the DCTV filter box, too. In addition, the DCTV software library must be present on your system - it's installed when you set up DCTV for use.

The settings in this requester don't change the screen on which Aladdin 4D's user interface appears. They only tell the program what resolution you want pictures rendered in. You can change these settings any time - useful in making quick tests.

Other settings in this tab

Aspect Ratio is the ratio of height to width for the chosen display. For Amiga display types on a "standard" Amiga monitor, leave at its default 0.868. After you've rendered a few circles, globes, and cubes (hint), you can adjust this factor as necessary to get the results you expect. This issue is discussed much more extensively later in this manual.

There are other check boxes in this panel, but we'll leave an explanation of those for later, too. For now, leave them at the default settings: Palette Match (ON), Dither (off), DCTV Filter (off but see above), and Frame Script (off).

24-bit Display Boards

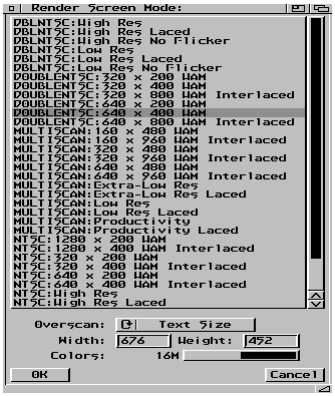
If you are using a 24-bit display board, do this in this order: Select an Overscan mode of your choice. Select Hires, Lace to your liking. Select the board's name from the top section. If you select the board gadget first, you have to manually adjust the others. Once you have the board's gadget selected, you can, of course, set the display size to your choosing. Most boards ignore the hires/lace gadgets and decide their output mode from the sizes entered.

Aladdin 4D directly supports Amiga AGA, DCTV, VideoToaster, OpalVision, Resolver, Retina, and all CyberGraphX (and compatible) display boards. If you have some other display card - and there aren't very many others - your card might still be able to display Aladdin 4D's 24-bit pictures if you render to disk and then use the board's normal software to display the resulting files on screen. FireCracker cards, for example, are only a display "buffer" and should be used with Aladdin 4D in this way. Aladdin 4D supports rendering directly to CyberGraphX screen modes. (Note: Don't use HAM.)

/// FLAG: trademark credits for verso /j

When loading drawings, the Editor's screen mode and aspect ratio are not loaded, but Preview and Render settings for these are. The Render menu item Render Screen also gives access to the screen mode for Rendering, but without the additional options just discussed.

AGA displays



If you're using an AGA machine, and either don't have or don't wish to render to a 24-bit display card, move the slider so that "Ham-8" is selected. This will give your picture a large palette, so transitions will be smooth around curves and in shaded areas. Of course, you can use any of the non-AGA modes, too, but Ham-8 makes great pictures and animations.

Using DCTV

DCTV, by Digital Creations, is an external hardware display enhancer that re-purposes the Amiga's Interlaced and High-Res display modes to deliver pictures in a large palette similar to television display. Aladdin 4D can render directly to DCTV's display modes. If you are using DCTV, do this IN THIS ORDER:

Select Overscan.
Select Hires/Lace.

Select the DCTV item in the display mode list window. Selecting the gadgets while in standard mode allows the gadgets to change the screen resolution for you. If you select DCTV first, you have to manually adjust the others.

Whichever device or display mode you chosen, accept the screen definition by clicking "OK".

DISPLAY BOARD USERS NOTE:

Many third party boards require an NTSC or PAL screen or a MultiSync monitor to function properly. Make sure you have the proper screen for these boards and set them up and get them working properly before attempting to write to them with Aladdin 4D.

OpalVision boards impose a maximum resolution that could be different from what you render. Aladdin 4D puts no limits on the size you can enter, but the board may not render or hold the information. This results in wasted render time. If you set up a size of 768x482 and your board is only capable of 736x476, you'll be calculating the full image and sending it to the board, which will quietly ignore the extra picture information and deliver only what it can do. Find out the maximum resolution of your board and limit render size to it. In 24 bits, even a little bit of extra picture rendering can take quite a long time.

Working in PAL modes

Aladdin 4D can work in any available Amiga screen display ~~mode which~~mode that meets your needs, including PAL modes. Just choose the mode you want for the Editor, Preview, and Render screens, as appropriate.

Setting up Preview modes

In addition to the Render screen's settings, Aladdin 4D lets you set up a preview screen to your liking. During creation of your drawings, especially if you wish to create animations, Preview offers a quick way to check your work. Previews are always one bitplane - that is, black and white, no grays - and wireframe only, no shading, shadows, nor other rendered elements (like textures).

You can change the Preview screen's specification, to suit the target hardware and display you want.

Choose the Render menu item "Preview Screen Mode." You're presented with a list of the currently available monitor types. Choose one; as with the Render screen setting, and choose an overscan type. You could also type numbers into the boxes, but they'll be filled in as appropriate to the mode and overscan setting you select.

The preview screen is limited to two colors regardless of the monitor type in use. This is both for memory and speed considerations.

Remember, when reloading saved drawings, the Editor's screen mode and settings are not loaded, but Preview and Render screen modes and settings are. When you save a drawing, Aladdin 4D records the screen mode settings for Preview and Render, but not the editor itself. You can change any of these at any time, of course.

Render Screen Options

Now that you have selected a display mode for Aladdin 4D to use, move the mouse pointer up to the menu bar and use the right mouse button (RMB) to select the Render menu item named "Render." You'll see a requester containing a ~~slider which selects the frame~~slider that selects the frame you wish to render and specifies other things for the program to do. For now, just hit <Return> to select the first frame. The Editor screen will close, and the render screen you have chosen will open. (If you're using DCTV, you may have to switch your monitor to view DCTV's output; if you use separate monitors, you should see the display there without any switching.)

The Amiga's friendly "Wait" pointer lets you know the program is calculating. Soon you'll see the image begin to appear as the program renders it top to bottom, left to right. Areas of the image that do not have any objects in them are being filled with a background image that was loaded from the disk. When the cube comes up for rendering, you will see that it has textures applied to it. Doesn't look as plain as the wireframe object that defines it, does it?

When you selected the default render, you told Aladdin 4D not to save the rendered picture. So, when after the screen has rendered, you can look at it, but you can't do much to save the picture. To record a rendered picture for posterity, select one of the save options in the Render/Render menu. Aladdin 4D supports all Amiga file formats, plus Toaster Framestore and formats to work with all CyberGraphiX, OpalVision, Retina, and FireCracker display cards.

If you're rendering and saving with OpalVision, Retina, FireCracker, or other supported 24-bit board, you do not need to save in 24-bit mode. The normal IFF save will automatically save a full 24-bit image from these boards. If you choose Save 24 Bit from the requester when using these boards, you'll actually be saving two 24-bit files, one from the board and one from Aladdin 4D's buffers.

Although Aladdin 4D supports rendering directly to a CyberGraphX screen mode, you should not use HAM with these display cards.

For practice, re-render the image of the cube to a saved format, to get the hang of saving pictures from Aladdin 4D.

Animation Quickstart

Besides rendering a single frame to almost any resolution, Aladdin 4D can create sequences of frames for use in building animations that will make the objects move and morph over time. That's the fourth "D": length, width, depth, and... TIME, for animation. Aladdin 4D can also save directly to animation formats in a variety of resolutions and file structures. You create these sequences of frames with items in the render menu, just as for a single frame.

The program creates preview animations in standard Amiga formats, useful for quick previewing of motion timings, textures, and the like. These animations can be played back in real time on your Amiga, or on other computers if you move the files over (and perhaps translate them to some palatable format).

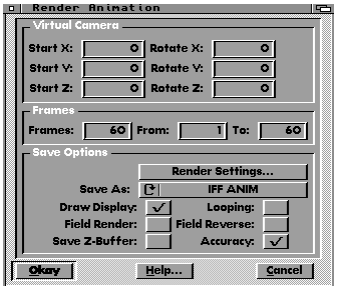
Animation rendering is controlled by the Render menu's Render Animation item. In addition to rendering for real, however, you can also render a Preview animation - controlled by the Preview Anim item. Both of these call the same requester: Render Animation.

This requester offers the opportunity to set up some parameters having to do with the animation to be rendered, including settings for the Virtual Camera (The one used if the drawing doesn't have one of its own, installed as an object), the number of frames to be rendered and which ones, and the method and format for saving the rendered frames and/or animation file.

The button labeled "Render Settings" offers a shortcut to the settings for rendering. It calls up the same tabbed requester as using the pull-down menu item Render / Render Settings.

Once Aladdin 4D saves the frames, Framestores, or other files for your animation, you can assemble and edit animations in a variety of ways, including single-frame video or film recording. You can assemble animations from frames created by Aladdin 4D using any Amiga (or non-Amiga, even) animation program that allows you to append frames (or fields) to an animation file, or which will single-frame record them individually to film or videotape. In addition, you can use Aladdin 4D itself to create Amiga-format animations in a variety of file formats for use by Amiga video packages such as NewTek's Video Toaster/Flyer.

Aladdin 4D includes extensive and extremely powerful animation features. Indeed, there is little about the animation process that isn't under your direct control.

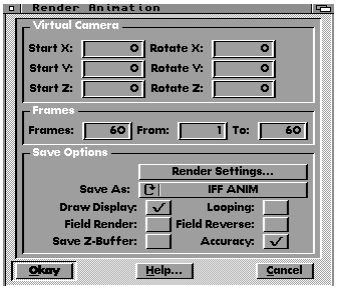


You can display and edit Amiga animations with any of the freely distributable or commercial viewers and other animation-friendly programs. If you're using a 24-bit board, you'll need to save the animation in a format that your board supports. Generally, a standard Amiga HAM or Ham-8 screen mode will work on these boards, but some boards will also display animation in much higher resolutions.

Most AGA-equipped Amigas will display Ham-8 animations, within microprocessor and memory limitations. These animations are sequences of high resolution screens in an advanced "Hold and Modify" display mode which provides a large palette. In many cases, these Ham-8 animations are indistinguishable from full motion video, especially in overscan display modes.

If you save an animation in the board mode for your 24-bit display, it will be saved as single IFF24 (24-bit) frames numbered sequentially. This can consume quite a lot of disk space, and you won't be able to view the final animation until you single frame it to tape or assemble the frames into an animation using software.) These animations are, however, of the highest quality you can get.

Previewing Animation



The Preview Anim feature in Aladdin 4D lets you take a peek at the way the program will create your animation before you devote time to rendering it. It's a good way to get a feel for the basics of animation, as well. You can save the Preview animations for further study, too.

Creating Animation

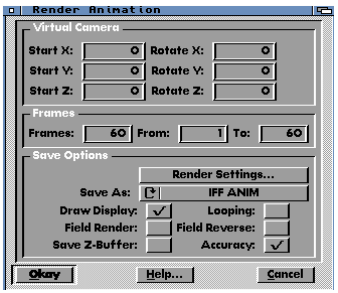
To create animation, set an appropriate screen mode from the requester brought up by the Render / Render Screen pull-down menu. If you are using a DCTV display, just change width to 320 and height to 200.

If you are in a standard Amiga HAM mode, change width to 160 and height to 100. Make sure the Colors slider reads "HAM" (or "Ham-8" for AGA). If it doesn't, change it using the slider.

These changes allow you to save "letter boxed" animation. The program will still use the screen size you set (the Hires, Lace and Overscan gadgets). All polygons, backgrounds, etc., will be scaled to this size. You may omit the HIRES flag unless you're using DCTV, which requires it.

Click OK, and you will see the render screen reset and begin showing the image in the smaller format.

Sequences of Frames



Select the Render menu item Render Animation. The requester that opens has boxes where you can enter a variety of information to control the type and size of animation that Aladdin 4D generates. More about this requester in the reference section of this manual. For now, the center panel is where we'll enter the number of frames that will be saved. The default is 120 frames which at 30 fps video playback speed is four seconds of animation.

The top panel is "Virtual Camera" which tells the program to rotate the "camera" around the cube (and everything else that might be present) during the animation. The contents of these boxes are saved with the project file, so if you alter them and then re-save the project, your new entries will reappear on re-load. If you mess these up really badly, reinstall only the sample files from the Install disks.

In the bottom panel, Aladdin 4D takes in information about how you want the animation rendered and saved. A ~~rotary gadget~~cycle gadget offers options on how you want the frames rendered and saved. To get multiple frames in a sequence, saved as individual files on disk, choose Individual 24-bit ILBMs, Individual Framestores, or Individual ILBMs with this gadget. The only other possibility is IFF ANIM, which creates a one-piece animation in the Amiga display mode you've established using the Render Settings menu. These settings are accessible from the Render Animation requester by clicking the Render Settings button in the bottom panel.

Accept the requester and you will be asked for a path and filename for the animation. Choose a good place to store the animation as it is being created by entering a path and filename in the requester. ~~Remember,~~Remember that such files can be huge, so be sure to choose a path on a hard disk drive with plenty of empty space.

As a guide, figure about a megabyte of space per frame when saving IFF 24-bit individual frames. Framestores are generally only slightly smaller, if any. It is not a good idea in general to use floppies for

animation storage, because it is so easy to overfill them, at which point the animation is lost. For this animation, with the small image size, the final animation will be around 400,000 bytes, so store to floppy if you like. How much room is needed depends on the size and type of animation you save, as well as the number of frames and how "busy" each frame is.

As the program calculates each frame, it draws it on the screen, or into the 24-bit buffer (or DCTV) you've selected. When each frame is finished, it is added to the animation file, then an estimated remaining rendering time is printed to the screen along with the number of the last frame completed. The time it takes to render the entire animation will depend on your system. It will, in general take two to four times as long on a 68030 as a 68040, four to eight times as long on a 68020 as a 68040, and 16 to 32 times as long on a 68000 as on a 68040. Of course, if you're running other programs in the background on your Amiga, the rendering time will vary with the amount of microprocessor time the other programs are using.

What you see on the animating cube as its frames are drawn is a digitized wood grain (oak) texture, fading away and being replaced with a bump map of noise with a little bit of color in it. Meanwhile, the Helix texture is becoming wider. All the time, the cube is being rotated, as specified. At the halfway point in the animations, the textures begin to reverse the changes toward their starting condition so that the animation will loop for continuous viewing.

If for any reason you wish to stop saving the animation, just press <Esc>. The key will not be honored until the current scan line has completed. When it's detected, you'll get a requester asking you to verify that you want to quit the animation. If you quit, the animation file itself is not deleted, but closed at the point where you stopped the processing. If you've saved three or more frames, the file will be a valid Opcode 5 animation, and can be loaded and viewed external to Aladdin 4D. You can use this stop-and-view feature to save time in preparing animations by viewing and saving only parts of an animation at a time. The saved parts can then be combined into a complete animation using any Amiga animation-capable editing program.

Previewing animation

The Preview Anim feature in Aladdin 4D lets you take a peek at the way the program will create your animation before you devote time to rendering it. It's a good way to get a feel for the basics of animation. as well. You can save the Preview animations for further study, too.

Use Aladdin 4D's Preview Anim feature to get a good idea of how your animation will look before you devote the computer time necessary to render it out in full resolution. The Preview Anim is a wireframe version of the objects, in motion. Save it as an Anim file and view it with your favorite Amiga animation viewing program.

Animation viewing

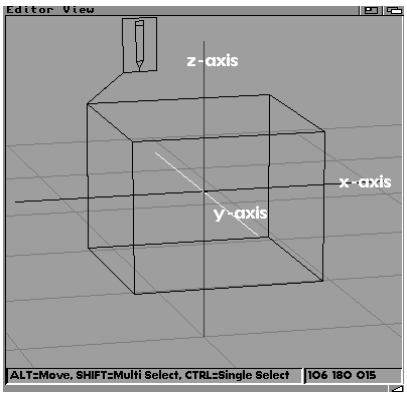
If you have plenty of memory in your Amiga, you can just leave Aladdin 4D running while you view the animation. Use the Front/Back gadget in Aladdin 4D's menu bar to push Aladdin 4D's screen to the background. The keyboard shortcut <Left Amiga> <M> does the same thing. Keep moving the screens to the back till you see the Workbench, where you can use your animation viewer to load the animation. If you do not have enough memory to hold it all, you must quit Aladdin 4D before loading the animation viewer of your choice.

After you have viewed the animation, let's look at the Editor itself for a quick overview, and at textures to see how the cube's effects were achieved.

Go back to Aladdin 4D and return to the Editor (or restart the program and reload the drawing if you have quit).

Introducing Aladdin 4D

Think of Aladdin 4D's Editor window as looking into 3D space, as though by putting your hand into the monitor you could reach around and touch the backs of the objects. In this window, polygons define the faces of objects represented on the screen in "wireframe," rather than solids, so you can see through them, though you can also see that they're three-dimensional shapes. The faces of the cube, for example, are square boxes, not solids. You can see other parts of the object through them, even though in the final rendered picture the faces will be opaque (or glass, or whatever). The vertical and horizontal dimensions on the screen are the X and Y axes of the 3D space represented. The X axis is the black one, the Y axis is white, and the Z axis is red. The default view is to have the Z axis vertically on the screen, with the negative end of the Z axis pointing "up". The point where these cross is affectionately known as "the origin." It basically means the center of this virtual universe.



This is like looking at a piece of paper lying on your desk: X running left to right, Y running to the front and back of the desktop and Z running down into the desk (positive) and up toward the ceiling (negative). This system is one of the standard mathematical and engineering representations of 3D space, and works well for our purposes. (If you use another 3D program, its axes' orientation may be different, but exchanging its objects is a simple matter of rotating the objects after loading them.)

The space is divided into arbitrary units called... "units." If you prefer, you can use "real world" coordinates - meters, that is. The conversion factor is 10000 units equal 1 meter.

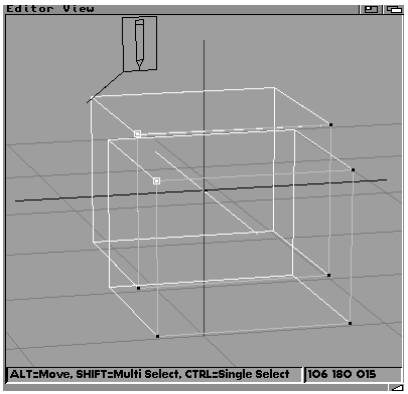
To change the type of coordinates in use see the Editor Prefs window. As you use the program, you can have coordinates - your choice - displayed in the screen title bar. The Coordinates option in the Settings menu toggles the currently selected point's coordinates to be displayed in the screen title bar. The editor window has a constantly updated display at the bottom that shows where you're pointing, moving, etc., as well as messages regarding the program's operation.

Other user preferences for the Editor window can be established with the Settings / Editor Settings menu item. See the reference section for a thorough discussion of those controls.

Aladdin 4D is a point based program. You select polygons by selecting any of the points which define the polygon. Every Aladdin 4D object is made of polygons, large or small, few or many. Place your mouse pointer on one of the corner points of the cube and click the left mouse button (LMB). If you have placed the mouse pointer close enough to any point (there's about a 4-pixel cushion) you'll see the polygons of the cube turn white. This indicates they have been selected. Notice that one of the polygons has red segments in it. This is the polygon that was selected first. The alternating red and white colors to the sides of the polygon are an aid in helping you distinguish where the sides are in the polygon. This is not very important in a simple square, but it's very useful with more complicated shapes.

Let's move the cube. It's still selected - you can tell because it's still white. Move the mouse pointer someplace. It does not have to be on the point once the ~~polys~~polygons are selected. Hold down the <Alt> key, and, while pressing the left mouse button, drag the mouse. The cube moves in space, relative to the direction of the mouse movement. The mouse pointer is blanked while you're moving objects, to reduce confusion about the relationship between the pointer and the object(s) being moved. In a 3D space such as this one, the mouse pointer doesn't give much of a visual clue, since it's suited more to a 2D environment. Try moving the object over to the right, then let go of the left mouse button. Move the

pointer to about the middle of the window and press the left mouse button again. Then move it to the right.



Note: You must keep the mouse pointer in the Editor window, out of the toolbox, and out of the menu area, as those other areas are reserved for other purposes.

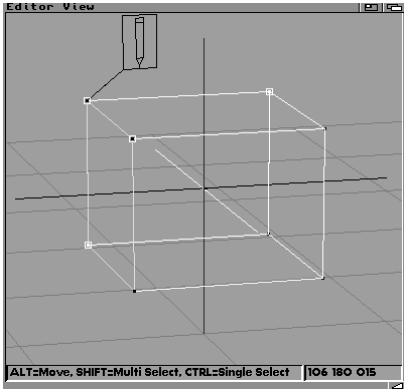
Still moving the object around? Press <Esc>. You'll see the cube resume its original position and turn black again, meaning it's no longer selected. The <Esc> key operates somewhat like an "UNDO" for Aladdin 4D. It actually only aborts the activity in progress and can only be used for the change you are currently making. Once you have "set" the change, <Esc> will have no effect.

When you clicked one of the polygons of the cube, they all selected. This is because they are grouped (in the current Group Level). If you only want to select one of the polygons in a group, hold down the <Ctrl> key when selecting a polygon's point. Try this out, to get a feel for the way this all works. Select one of the polygons then press <Esc>, then another, etc. Go ahead and move them (hold down the <Alt> key) to verify that only one polygon is selected.

Move to First

You can only select certain polygons. This is because when you select a point in the cube, the program selects the first polygon it finds in its list of polygons. It maintains this list in the order in which polygons were made. You can move polygons around in the list using the Edit menu items "Move to First" and "Move to Last". If you select a point and the face that becomes selected is not the one you want, invoke the "Move to Last" menu item. After it changes back to black, try clicking the same point again. Another face will select, because it is now closest to the first polygon in the list.

Multiple selection



To select two or three of the polygons in the cube, even though they are grouped, just hold down BOTH <Ctrl> and <Shift> while selecting. The <Ctrl> key says "select only one poly, even if grouped" and <Shift> says "select another poly even if one is already selected." Aladdin 4D provides other ways of multiple selection, but we'll come to those later on.

If you do not want to move the entire polygon, just one of its points, do not hold down the <Alt> key.

NOTE: If you move only one of a polygon's points, be sure to keep the polygon flat - do not move a point into a different X, Y, or Z plane, such that the polygon itself becomes three-dimensional, Polygons must remain flat to render properly. It is possible to correct the problem by converting ~~polys~~polygons to triangles, however. This issue is discussed more thoroughly in the section on degenerate ~~polys~~polygons.

If you move the cube, and decide that you're happy with its new location, issue the "Set" command to make the move permanent. This sets the point(s) that have been moved to their new position(s).

To set any action, click the right mouse button (RMB) once. When you set an action, the polygons go from selected to unselected in their new positions. The old positions are replaced with the new ones, and the undo buffers are loaded with the polygon's new positions. Try this out a few times for practice.

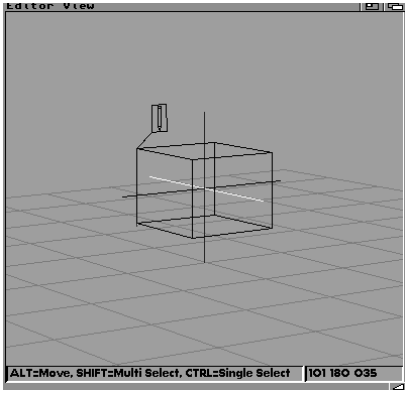
Attach Points

Do you see the funny looking marks on the screen at the location of the point you have selected after you set it? This is the Attach Point. It has many functions in the program, and you will find it very important in dealing with objects as you progress.

If you want to re-center the cube to the origin, there is a simple way. First make sure nothing is selected. Move the mouse pointer over the Page Move Gadget and click the right mouse button. This moves the Attach Point to the origin. Select the cube and click the right mouse button on the Page Center gadget. This centers all selected polygons to the Attach Point. This operation and its variations will be very useful in working with objects. If you like, try selecting a point on the cube, then using the Set command. The Attach Point will be at the point you selected. Then select the cube and center it to this point. If you use the right mouse button on the Page Move gadget while polygons are selected, it puts the Attach Point at the center of the selected polygons, not the origin, which as you can imagine is also quite useful.

If your Amiga mouse has three buttons, instead of two, you can use it to do what clicking Page Move does, without moving the mouse pointer over the gadget box. Holding down the middle mouse button pans the editor view (just like clicking the Page Move gadget in the Toolbar).

Alternate views



Often, it's not very informative to look at objects head-on. When dealing with three-dimensional shapes, it'd frequently be nice to get a look at them in perspective, or to take a look from several perspectives. The computer screen won't really let you reach your hand in, but Aladdin 4D is capable of twisting and turning its display so you can see things from the other end, or from the top, from the side, or from any corner.

How do you change the direction you are looking? When you started Aladdin 4D, the axes were in the center of the Editor window, lined up with the sides of the screen. To change this, use the <1>, to <9> keys on the number pad. The number keys at the top of the keyboard do different things - use the number pad to change the view. The <1>, <2>, and <3> keys control rotation around the Z axis, the <4>, <5>, <6> keys control rotation around the Y axis, and the <7>, <8>, and <9> keys control rotation around the X axis.

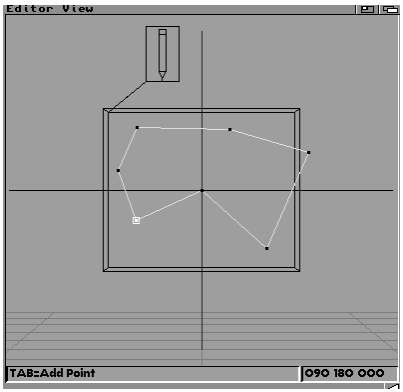
If you have lots of ~~polys~~polygons in the drawing, when you rotate your view (and use some tools) the program will paint only partial polygons, for speed. You can toggle this with the Settings menu item "Auto Quick." It is on by default - that is, checkmarked in the pull-down menu.

Three special "flat" views are used for freehand drawing, scaling, etc. To see these, select the Z Active Axis (see the gadget layout), then select the Flat View gadget or tap the <Spacebar>. You will be looking at the drawing from the negative end of the Z axis. Try this with the X active axis, and finally the Y.

The keyboard shortcut for the Flat View gadget is <Spacebar>.

While in the Y Active Axis, Flat View, let's try a Freehand Polygon. Issue the "set" command (RMB in the Editor window). Now select the Free Hand Poly gadget with the left mouse button. The mouse pointer will turn into a pencil. Move the mouse pointer into the window and hold the left mouse button down while dragging the mouse. You'll see a line being drawing itself in space. Once again, this line is not "attached" to the mouse pointer.

The new polygon always starts at the Attach Point. Pull the line out slightly, tap the <TAB> key once, then move the line some more. <TAB> creates a new point. The polygon now has three points. Do this several times, making a polygon of quite a few number of points. When you are done, do not use the <TAB> key to finish; after positioning the last point, use the Set command (RMB with pointer in Editor window). This will deselect the poly and reset the undo buffers for it.



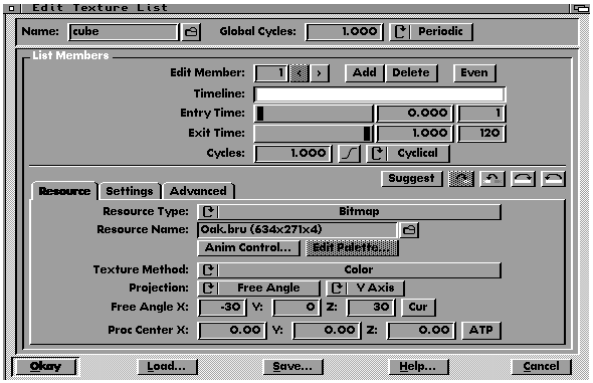
You may want to get back to the original view direction. Select the Render menu item Restore View. The view returns to the original view contained in the drawing when it was loaded. Did you notice the Record View item? ? It does what its name implies: records the view position so you can restore it when you want to have another look from the same angle. Use Record View and Restore View as you need, to make looking at your object from different views easy.

Texture Quickstart

The cube's textures…

So how did the cube get its textures? ? This issue is too advanced for a quick start, but let's take a cursory look. Aladdin 4D has extensive (and that means extensive!) texturing abilities. As with all of its other features, the emphasis in texturing is on user control. Set the polygons that you've just created, and let's take a quick tour of textures.

To begin, select the Object menu item Textures. This opens the Texture List Selection requester. The items listed in the center panel are the textures that are in use on all of the objects in the Editor window. Click the one called "cube," then click the Edit gadget. Another requester opens, this time the Edit Texture List requester.



The cube's texture is listed in the Name box at the top left of this requester. You can change the texture to work on without returning to the previous requester - just click the button next to the Name box to get a list of available textures.

This requester is where you apply and set up textures. Texturing in 3D is a complex subject, so the control requester is extensive. Tabs across the bottom of the window group the various controls into logical units. The first of these is the Resource tab. This part of the requester defines the name, type, and size of the bitmap used for this member. The cube has the oak woodgrain texture applied, so its name appears in the Resource Name box. The gadget to the right of that box lets you change this by loading a new texture from disk.

There much other information here that we'll come back to later. For now, look at the gadget labeled Edit Member. It reads "1". Change this to 2.

If you watched closely, you noticed that many of the gadgets changed. This is the second texture to be applied to the same object - remember the woodgrain changes during the animation. With Texture Number at 2, the parameters being displayed pertain to a "procedural" texture called Noise Closed.

The next tab, moving right along now, is Settings. This contains sliders for Color and Strength, providing entry and exit values for those elements of the texture. Throughout Aladdin 4D's menus and requesters, Entry and Exit values refer to the values and beginning and end, respectively, of an animated sequence. The Noise member of the Texture List is set to zero strength at the beginning of the animation and full strength some time later - begin and exit - with the program averaging the strength into a smooth transition during the course of the animation.

Change the Member Number to 3. This member of the Texture List is the procedural Helix texture. The program adjusts the content of the entry boxes according to the type of texture you're working on. Items that apply to bitmap textures don't necessarily apply to procedurals, and vice versa, for example.

Applied textures are a very important tool in the 3D rendering and animation workshop. This quick tour leaves much of the power of textures untouched, but it gives you a brief non-intimidating look at the Aladdin 4D's texturing interface.

Quick Preview Mode

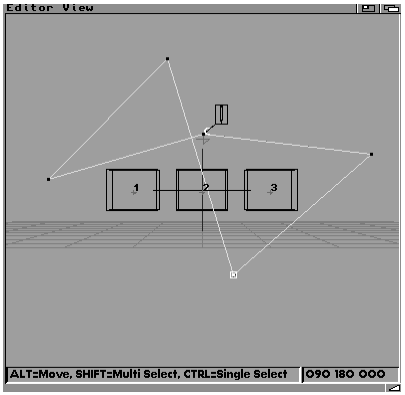
Before leaving this drawing let's take another quick peek at Aladdin 4D's Preview mode. Close the texture requesters to get back to the Editor. When you rendered the animation, the cube rotated around the Z axis 360 degrees. Press <F9>. What you see is Aladdin 4D's Quick Preview mode. This mode is interactive. Hold down the right mouse button to temporarily stop the animation so you can inspect the position of objects, etc. To close Preview and reopen the Editor window, hit <Esc>.

Your choice of screen mode for the Preview Screen is established with Preview Screen Mode item under the Render pull-down menu. The Editor's screen mode can be set with the Editor Screen item in the Settings menu.

Animation Preview is interactive, very much like the Editor, if you do not have a camera in the space being previewed. Aladdin 4D supplies a default camera that watches the keyboard. You can add cameras to your drawings with the Object menu item "Camera." If you have added a camera, Preview uses it instead of the Editor's virtual camera, and it's intelligent about animation sequence frames used as textures. While in the interactive preview, the up/down cursor keys zoom in and out, and a tap on <spacebar> restores the view in Preview to what it was when you started it going. Holding <Shift> while using the cursor keys moves the viewpoint vertically, and horizontally. Unshifted, <Cursor Left> and <Cursor Right> increase and decrease the angle of view.

Before you press <Esc> to return to the Editor, press <Spacebar> to restore the view to what it was when you first entered the preview. If you return to the Editor without doing this, the changes you make in the view direction will carry over to the Editor.

Enough cube, already



This ol' cube is getting, well, boring, so.. let's look around a little. There are actually two drawings in the cube's drawing file. The first one defines the objects, as you've seen, and the second one is in a completely different space that you can't see unless you jump to it. Look at the bottom of the gadget tool box. At the moment, the bottommost left-hand gadget has "1" in it. Click the right-arrow gadget to its right, and "1" changes to "2". A different bunch of objects appears in the Editor window, too. You have jumped into the drawing's second "Space." The first space is still there, but you've moved. You can return to it any time by clicking the Left arrow gadget next to the "2" (or just edit the box contents to read "1", whichever.

This new space contains objects similar to the first, except there are three cubes and some funny looking red polygons with numbers on them. One has a "C" on it. This is the camera. It is a degenerate poly (one that can't be rendered) so it is (should be!) unique looking. The other three, with numbers, are targets. The big, bent polygon is a path that the camera has been assigned to follow. Preview this animation using <F9>. Use the Render Anim menu to set up about 120 frames of animation if it's not already set.

The keys only affect the "virtual" camera, so do not use the keys in this preview. They will have no effect on your view, as the camera object has taken control. The keys will, however, alter your Editor's view, so when you return you might not recognize where you are.

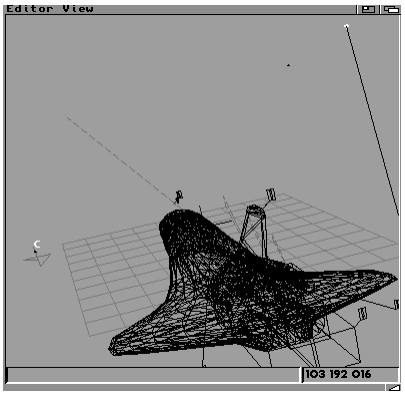
What you see in Preview is the camera's view as it roams around the bent polygon. The three targets take control of the camera, one after the other, telling the camera where to look and what zoom to use. The targets get centered in the view one after the other, with smooth transition to the next target. Return to the Editor (<Esc>). Select the rightmost point in the big path, and move it to a new location (Don't use the <Alt> key; only one point moves), much higher on the screen. You might want to zoom the Editor's display out (<Cursor Up>) to give you more area to view.

When you have moved it use set the point to its new location (RMB), and preview the animation again, and you will see that the camera follows the new path you've specified. Try some other positions for the points, maybe flat viewed in the Z axis. The cubes are clones of the one in Space #1. If you wish, you can render the animation to get accustomed to the way camera paths work.

Quick Tour 2: Buy the “T-shirt”

Whew!-! Our "quick" tour turned out to be not so quick, didn't it? But it's done, now, so that's it for your first experience with Aladdin 4D-. Drop by the lemonade stand on your way out, and be sure to buy the souvenir "Aladdin 4D Quick Tour" T-shirt-. In this visit, you've seen some of the program's power and flexibility, and learned to work some of its controls-. There is much, much more-. Aladdin 4D can bring some of your most abstract ideas to (virtual) life.

Okay, so much for boring cubes-. It's time for something more complicated: a space shuttle-. The next section (Quick Start 2: The Sequel) has a complete space shuttle object to work on-. With some effort, you can "fly" the shuttle out to Mars or one of the comets and back, and maybe bring back more of those funny rocks that has the scientific world abuzz-. This object was originally distributed with an earlier version of Aladdin 4D.
~~For this release, it has been updated it to use some of the more advanced features that are now available.~~



The shuttle object will demonstrate an advanced drawing, to give you a better idea of what you can do with some of Aladdin 4D's many features-. This drawing uses some of the program's most memory intensive features, like full 24-bit textures, and Flare objects-. You'll need around six megabytes of free memory on your system to load and render this drawing-. If you run into erratic behavior when looking at this drawing, it is probably because you don't have enough available memory on your Amiga.

If you're running a machine with low memory, do not despair-. It is not necessary to see this drawing render to learn how to use Aladdin 4D-. You can just load the finished, rendered image from disk to follow what the discussion is about.

Shuttle

If you have enough ram on your machine, Open the drawing named "shuttle.4d." As it loads, you'll see many textures being loaded-. There are multiple defined spaces as well-. As you read along, don't worry if you don't fully understand each step in this discussion, yet-. After you've worked through the tutorials and reference sections, all of this will be easily understood.

After the shuttle drawing is loaded, you will ssee the space shuttle model in the Editor-. Notice that you are in Space #2-. Move to Space #1 by clicking the little left arrow at the bottom of the tool box-. Here you see the template ~~polys~~polygons that were used in modeling the shuttle-. These templates were drawn by measuring a small plastic model of the shuttle-. The templates are "cross sections" of the shuttle's parts, like the formers you'd use to build a model plane or boat-. After these templates were drawn, the Extrude Tool was used - with its Poly To Poly option - to "skin" the templates-. Then the skin was converted to triangles to get rid of non-planar (not flat) ~~polys~~polygons that may have been created.

Textures for the model were hand drawn in a standard Amiga paint program. The basis for the paint images was bitmaps created in Aladdin 4D. The technique is to make "template" views of the model's main parts by not using a camera, decreasing the wide angle view to obtain as flat a view as possible, then rendering one-bitplane silhouettes of the main parts alone. These silhouettes were loaded into the paint program, to define the shapes that were to bear the textures. The textures were drawn, then loaded and applied to the objects in Aladdin 4D. Why not stop at the paint stage? The answer is, of course, that You'd use the animation program to make such an object move, rotate, and "fly," and let the 3D geometry and shading calculations perform the darkening and lightening as appropriate on the bitmapped textures.

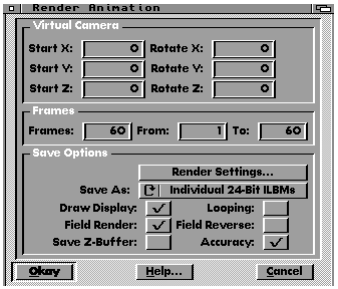
Back in Aladdin 4D, after making the rocket engines (with the Lathe tool applied to a hand drawn template poly) five gases were created to simulate the rocket flames. The gases use some noise that "rolls over," animating the flames.

The next step was adding Flare objects at the source of the rocket flares, with a few more thrown in for overly bright star objects. An appropriate motion was decided on, and the paths were drawn, lights added, shadows decided, and as a final step, a suitable background image defined.

As loaded, the display is set up for regular Amiga Ham mode - 4096 colors. This is not very satisfying, compared to some of the display enhancers available, and it's nowhere near the full capability of Aladdin 4D, but it will render in a minimal system and display on any Amiga with or without AGA chips and display boards. If you have a display enhancer or an AGA machine, go to Render Settings and set an appropriate display. In Render Settings, shadows are ON. For a faster render time, you could turn this off. The options for Phong shading, textures, transparency, background, etc., are also ON. This drawing must use all of these.

Once you have the display set up to your liking, render the first frame. If you like what you see, you can render the entire animation. Preview it, at least, to observe the motion. You may want to save the animation so you can play it back in real time on your Amiga.

Field Rendering



If you want to get really fancy, just to show off, turn on Field Rendering, which closely simulates live video capture. When Aladdin 4D renders fieldwise, object positions are recalculated 60 times per second, nominally, as opposed to 30. The resulting motion appears much smoother, and the "look" of the finished animation is closer to what real videotape would produce from live action.

When you capture video on your camcorder or tape deck, it records each frame as two fields, interlaced. Each field is a distinct image in time, half the vertical resolution of the picture, and ~~offset by one scan line~~offsets by one scan line from its other half! Motion carries on during fields, not just frames. This means that even though the equipment is making only 30 images per second, each image is composed of two distinct field images. These fields are held in the even and odd scan lines. The even scan lines are field one, and the odd scan lines have field two. When your playback equipment plays the video, it displays only 30 frames per second, but for each frame, two fields flash by, so you are actually watching 60 fields per second. This makes motion much smoother than would be possible at only 30 distinct time slices per second. By imitating the video capture, Aladdin 4D's field rendering can achieve the same smoothness of motion, eliminating the telltale computer animation jerkiness of mere 30 frame per second renders.

To use field rendering, you simply turn on the Field Render option in the Render menu's Render Animation requester. When the animation renders, you'll see the image being painted in two distinct fields. First the even scan lines will paint, then after a brief time, you'll see the odd scan lines interlace into the image, creating the complete frame. After the frame is complete, it will be saved. Aladdin 4D's method of field rendering takes only a small amount of extra time over regular framewise rendering.

To play back a field-rendered animation and see the field render effects properly, you MUST play it back at exactly 30 frames per second. Also the animation MUST be in an interlaced screen mode to duplicate the conditions of the playback equipment. Field rendering is really meant for videotape playback, but you can still see the improvements if the anim is interlaced and plays back at exactly 30 fps.

Oh, there's one potential gotcha. Some video equipment plays field 2 first, then field 1. If you have that type of equipment, just check the Field Reverse button before rendering. For most equipment, you can ignore this option.

TUTORIAL SECTION

Tutorial (Drawing)

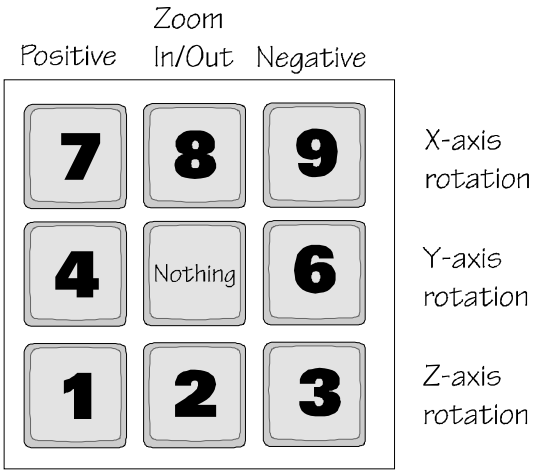
We trust you're back from the remote galaxies, parked your shuttle cozily and didn't forget to drain the radiator. Welcome to the Aladdin 4D tutorials section. Pull up a comfy chair, get your mouse finger at the ready, and let's get started. These tutorials are designed to give you maximum information about operating Aladdin 4D, and in the minimum of space and time. As with the first part of this manual, it's assumed that you already know the basic operation of the Amiga, such as running programs, menu selection, gadget operation, paths and filenames. To that knowledge, you've added a basic familiarity with Aladdin 4D: loading and saving drawings, operating some of the menus, making and setting rudimentary objects and moving them around, etc.

The Reference sections indicate the location of gadgets, so this section will just say "click the <whatever> gadget", without belaboring where to look for it. Many of the tools are actually small programs offering many options and diverse abilities. For clarity, the various settings are mentioned in general. Refer to the Reference section for further details and clarification.

Aladdin 4D starts (open it now) with its Editor Screen, comprising the menu bar at the top, the floating toolboxes of gadgets, and the Editor window occupying most of the screen. You can resize the windows, including the Editor window, to your liking. If the toolboxes are smaller than the material they contain, proportional sliders will indicate there is more to see.

Both the internal and the external toolboxes are movable, closeable windows. To close a window, click its close gadget in the upper left corner of the window. To reopen it, use the corresponding Settings menu item to return it to view. You cannot close the Editor window - it has no close gadget - without closing the whole program (Select Quit from the Project menu or its keyboard equivalent, <Amiga> <Q>).

Aladdin 4D remembers the last position and open/closed state of the tool windows, even after you shut off the computer and restart the program cold. It also remembers the last screen modes and aspect ratios that were in use. When loading drawings, the Editor's screen mode and settings are not loaded, but the settings for Preview and Render are.



Tool position lock

You can lock your favorite tools in position in the Tool windows. If you move the mouse pointer over a tool, but all the way to the right side of the gadget area and then click the RIGHT mouse button, the tool will jump to a "permanent" position at the top of the scrollable list of tools. It's marked with a double less-than sign (<<) so you can tell it's locked. If you scroll the tools, the locked ones will not move. Lock a couple more, and you will see that locked tools are also put in alphabetical order. To unlock a tool, just use the same procedure - right mouse button at the far right.

When you quit, Aladdin 4D remembers the state of the locks and opens them in the same state on your next session. This is independent of any particular drawing.

View Angles And Magnification

The angle at which you view the drawing in the project can change, depending on your needs. While it's convenient to work from the side on many objects, some might be easier to construct from the top, or even from a perspective view. Indeed, you can switch from one view to another, or "nudge" the view in any direction, for a better view of what you're working on. You control the view direction with the number keys in the keypad. If yours is one of the Amiga models that don't have a keypad, the regular number keys will work. On Amigas with keypads, the number keys at the top of the keyboard change the Space indicator. You can also just manually enter the view direction. The number keys make up a little matrix of three rows and three columns.

Row three (<1>, <3>) controls rotation of the view around the Z axis.

Row two (<4>, <6>) controls the rotation around the Y axis.

Row one (<7>, <9>) controls the rotation around the X axis.

For all of these, column <1>, <4>, <7> rotates negative, and column <3>, <6>, <9> rotates positive. The center column (<3>, <8>) zooms the view in or out; <5> does nothing.

In addition to the view angle, you can select the magnification applied to the view. This is the "zoom" feature, applied with the Up/Down cursor keys, and also implemented on the <2> and <8> keys of the Amiga number pad. Practice using these keys until you are comfortable with them.

Note: to restore the view to the "default" view the program opens in, just press <Spacebar> any time. This lets you get back to "ground zero" without reversing all the gyrations you've applied with the view controls.

Press the right mouse button over the Flat View gadget to open the View Angles requester, where you can directly specify the angle of view. The current angle of view is printed in the bottom right corner of the Editor.

Aladdin 4D uses three "primary" views. These are "flat", looking directly down from the negative end of any one of the axes. You get a Flat View by selecting the Active Axis gadget of your choice and clicking on the Flat View gadget. Do this for X, Y, and Z, to see how it works. This is important. You'll be asked repeatedly throughout this manual to select "Flat View in the Y". This means to select Y as the Active Axis and click the Flat View gadget.

You can control the magnification of the Editor or Preview view with the Amiga's cursor keys. Tap <Cursor Up> a few times, and you'll see the axes get smaller. <Cursor Down> makes the axes larger. You can also use the Zoom gadgets to either double or halve the magnification. By selecting the Zoom gadgets with the RIGHT mouse button, you get a requester that lets you directly enter the magnification you want.

While working in the Editor window, you can generally double-click an object to bring up the appropriate edit requester.

Page Position

To move the view left, right, up, or down, use <Shift> and the cursor keys. Try this now. To re-center the view to the origin, click the Page Center gadget (LMB). You can also use the Page Move gadget. Click this and move the pointer into the Editor window and press and hold the left mouse button while moving the mouse. Try this. Once you have clicked the Page Move gadget you must move the view before continuing any other operation.

When moving the view position all perspective is maintained in a proper manner for the new view position.

Perspective

You can control the perspective of the view using the <Cursor Left> and <Cursor Right> keys. Try this. These keystrokes move the virtual camera closer and farther from the origin, while adjusting the zoom to compensate and maintain the image size. The virtual camera is not visible, but exists as a point in space from which you are observing. The <Cursor Left> key moves it closer, and <Cursor Right> key moves it away from the origin. There are limits to how close you should move the virtual camera. Basically you should not move the camera so close that there are polygons behind it. Open the Observer Position requester using the right mouse button on one of the Zoom gadgets. Zoom is the current magnification. Its default is 4000. The Camera Distance is the distance of the virtual camera from the origin. It defaults to 160000. Change these to the defaults and accept the requester.

Isometric View

Sometimes it's bard to conceptualize a complex object by looking at it head-on in the X, Y, or Z. For a better three-dimensional idea, you can see the drawing in a non-perspective, or isometric view. Click the Isometric gadget. In this view, all perspective is removed from the drawing. All lines that are parallel in space are parallel on the screen. This type of view is particularly useful for placing polygons relative to one another. Click the Isometric gadget again to return to the perspective view. Perspective views cannot be rendered, but are used in preview if only the virtual camera is present.

Ground Grid

A Ground Grid is available in the Editor to help you line things up precisely. You can turn this on or off as you need it. Like the axes, the Ground Grid is a virtual object and cannot be selected. The grid is turned on/off, and its parameters can be changed by selecting the Preferences menu item Attach Point Len/Ground Grid. There's more information about the Grid later, and it's described fully in the reference section.

Drawing Objects

There are many ways to draw in Aladdin 4D. You may use the Freehand Poly tool. You may make primitive arcs and rectangles. You may create primitives with some external tools. You may use the Freehand spline tool. You may even draw in another program like Professional Draw, print the drawing to an Encapsulated PostScript file (EPS), and load the EPS file into Aladdin 4D. Aladdin 4D also imports several types of files, including Videoscape .GEO and Lightwave object files. For this tutorial, we'll explore both of the freehand ways.

Drawing ~~Polys~~Polygons

First lets look at the Freehand poly method.

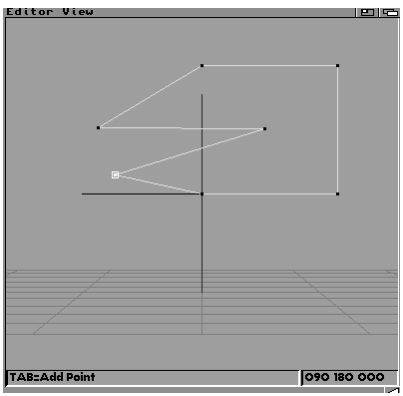
Flat view in the Y axis.~~.~~ When you draw freehand, the new polygon always starts at the current Attach Point.~~.~~ You can set the Attach Point with the Edit menu item Set Attach Point.~~.~~ This is not the common method, however.~~.~~ The common method is to use the mouse to set the Attach Point to any point on any poly in the drawing.~~.~~ When you start the program, the Attach Point is at the origin (0, 0, 0).~~.~~ Select the Freehand gadget with the left button.~~.~~ Move the pointer into the Editor window and hold down the button while moving the mouse.~~.~~ You'll see a line come from the origin.~~.~~ This is the first side of a new polygon.~~.~~ The first point (point 0) of the new poly is at the origin, and the second point (point 1) is at the end of the line, the one you are moving.~~.~~ The point is not attached to the mouse pointer.~~.~~ Release the mouse button and move the mouse pointer to another position in the Editor window,
then press the left button again, and move the point again.~~.~~ Observe how the mouse pointer is used as a reference for direction and as a quantifier only.~~.~~ This allows you to do two distinct things.~~.~~ First, you can move the point outside the Editor window and still control it.~~.~~ Try this.~~.~~ Second - and very important - you can use the bottom and left side of the screen as straight edges.~~.~~ This is a valuable feature that's you'll probably use often.

Tap the <TAB> key, then move the point.~~.~~ You have created a third point (point 2), and it is this new point that you are moving.~~.~~ You may create any number of points by tapping of the <TAB> key wherever you'd like a new one created.~~.~~ You should not create multiple points that are in the same location for a polygon you wish to render, although doing so is useful for paths. ~~The program paints the polygon in alternating red and white segments for clarity.~~

Setting ~~Polys~~Polygons

When you've created enough points, use the Set command.~~.~~ Do this by clicking the right mouse button when the mouse pointer is in the Editor window.~~.~~ The Set command does two things.~~.~~ It tells the program you are finished creating a new polygon in Freehand mode, and it resets the undo buffers for all polygons.~~.~~ The Set command is used very often while working in the Editor.~~.~~ When asked to "Set" ~~polys~~polygons, this means to move the mouse pointer into the Editor window and click the right mouse button.~~.~~ Use the Set command now to tell the program you are finished with the polygon you are drawing.~~.~~ You'll see it change to black, indicating that it's no longer selected.

Selecting ~~Polys~~Polygons



The polygon you have just made is unselected.~~.~~ To select it again any time, simply click any one of its points.~~.~~ Do this now.~~.~~ The polygon changes color when it is selected.~~.~~ You must select the polygon by one of its points.~~.~~ Aladdin 4D requires an active point, not just polygon, for all of its operations, as you'll see in most of the tools.

Adding and deleting points

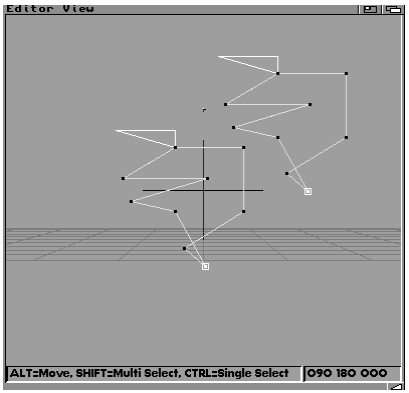
For many purposes, You may want to add or delete points to existing polygons.~~.~~ Click one of the points in the polygon, then use the Delete Point gadget (LMB).~~.~~ The point you selected is eliminated.~~.~~ Do this a few times, but make sure you leave at least two points in the polygon.

To add points, there are two methods. First, you can select an existing point, then click the RIGHT mouse button on the Delete Point gadget. This creates a new point for you at the same location as the selected point. You can then position the new point where you need it. Do this a few times to get the feel of it. The second way you can create new points is by using the <TAB> key. Select one of the points of the polygon and tap the <TAB> key. You will see that the mouse pointer changes into the pencil again, and you can move the new point. You are back in Freehand mode, and can use the <TAB> key to continue to make new points. Use the Set command when finished.

Moving polyspolygon/points

To move one of the points of the polygon, select it and then press and hold the left mouse button while moving the mouse. To move the entire polygon (actually all selected polygons), hold down the <Alt> key while moving the mouse. Try this with the polygon. Move it around awhile to get used to it. Now set the polygon.

Grouping polyspolygons



You can group and ungroup polygons in the drawing. Make two more polygons. First set the Attach Point to one of the points in the current polygon by selecting the point, then using the Set command. You should be able to see the Attach Point after the Set. Get the Freehand tool and make a polygon there with several points - at least four or five. Now move the Attach Point to one of the new polygon's points, and make a new polygon there with four or five points. You now have three polygons. Select one of them. Notice that only the one you chose becomes selected. Do not Set it. Instead, hold down the <Shift> key and select another of the polygons. Both polygons become selected.

To group the two polygons click the Group gadget (LMB). The polygons will Set, and both are now members of a group. Selecting either one of the polygons selects both. Move them, and both polyspolygons move. Set them, then select and move the third poly, and you'll see that it is independent of the other two. Set it. Select all three polygons and group them. Now when you select any polygon of the group they are all selected.

You will often want to select only one, or part of a group. There are two ways to do this. The first uses the keyboard. Set the polyspolygons, then hold down <Ctrl> and select one of them. Only the poly you choose becomes selected. It is still a member of the group, but the <Ctrl> key tells the program to ignore groups when selecting. If you want to select another polygon in the group, hold down <Ctrl> and <Shift> while selecting.

The <Ctrl> key tells the program to ignore groups when selecting, and <Shift> tells the program to allow another selection.

The second method uses a tool in the tool box. Set the polyspolygons. Select the Multiple Select gadget with the left mouse button (Using the right mouse button on this gadget will select all polygons in the drawing). Bring the mouse pointer into the Editor window. Imagine a box around one of the polygons. Move the pointer to the upper left corner of the imaginary box. Click the left mouse button. Move the mouse pointer to the lower right corner of the imaginary box. You'll see a "rubber band" box

form.. Click the right mouse button.. If you have the polygon completely inside the rubber band box, it will select.. You may select any number of polygons in this manner.. All polygons that are completely inside the rubber band box will select.

Aladdin 4D also permits "lasso" and "polywrap" selection methods.. This means you can rope in an irregularly shaped area, and select everything within the bounds of the shape you draw with the mouse.. These processes of selection ignores groups.. Use the right mouse button over the multiple select gadget to bring up a requester in which you can edit its mode of operation.. You can change this any time.

To ungroup polygons, select one of the three polyspolygons (use <Ctrl>) and then the Group gadget with the right mouse button.. The poly will set.. Now select the poly again.. The other two polyspolygons in the group no longer select.. The poly is not a member of their group.

Multiple Group Levels

Groups let you manipulate many polygons at once, saving much repeated (and boring) work.. However, for some purposes you might want your polygons grouped in one way, yet for other processes you might want them grouped differently.. If you had to un-group and re-group them each time, you'd perform lots of (boring) repetitious grouping actions.. To save time, and make things as friendly as possible, Aladdin 4D permits multiple group content definitions, so you can make some groups and keep their definitions around, then make others that don't necessarily include the first groups.. You can switch "group levels" anytime you like.. The left mouse button moves forward in the list, and the right mouse button moves back.. A few of the levels are reserved for special purposes, but there are seven in all.

Look at the area just to the right of the Group gadget.. This area displays a number (usually), which is the active, or current Group Level.. At this time it displays a "1".. Click this area with the leftright mouse button and it will display a "2".. The program now has an active second Group Level.. Select one of the grouped polyspolygons and notice that it is not grouped with any of the others, although it is still grouped with the other in Group Level 1.. You can group your polyspolygons in a different fashion within each Group Level.. Normally this is used to allow easy editing.. For instance, you may have a car, with the polygons making up individual parts (wheel, tire, door, trunk, hood, each fender, each headlight, etc.) as independent groups in Level 1.. In Level 2, however, all the tires might be grouped together - because you'll want to operate on them as a group when it comes time to animate them - and all the lights, the whole body and so on.. In Group Level 3, you might have the whole car grouped together.. This makes it convenient to select and change attributes and textures for certain parts, and also to select and move the entire car without having to select each part one by one.

Group levels

Five group levels are open for general use, and you can have more than 65,000 groups of polyspolygons in each level.. Two special groups (Tthey display "Sha", and "Sdw"" and “Sdw” instead of numbers) are for shading and shadows, and are discussed later, in their pertinent sections.

Groups may include Aladdin 4D's Procobjects.. These are special polygons and objects such as Flares, Fountains, and Gases.. Procobjects reload their groups related to normal polygons when loading a drawing.

/// FLAG: Procobject Groups 01/17 -/j

Hiding and showing polyspolygons

Often you might want to see (or render) only a single group or two, or even part of a group. Aladdin 4D has two ways to allow this. The first way is with the Hide/Show gadgets. Select the polygon that is not grouped. Click the Hide gadget with the left mouse button. The polygon is no longer visible. It isn't removed from your drawing, just made temporarily invisible in the Editor window. Now click the Show All gadget. It is now visible again. Select the poly again, and this time click the right mouse button on the Hide gadget. Notice that only the selected poly is visible. The right mouse button hides all polyspolygons except the one (or more) selected. Grouped polyspolygons are treated differently. Use <Ctrl> to choose only one of the group and hide it. Now you have one poly of the group visible and one hidden. Select the poly of the group that is still visible - the program will still select all members of a group, and if any are hidden, will reveal them. If you wanted to select only the one poly, you must still use <Ctrl>, even if the other members of the group are hidden.

The second way to "hide" polyspolygons is to use the Space controls and jump them to a new or existing space. See the section on Space control for further discussion of this method.

Selecting the Group Number is easy, The Group Number Selector gadget honors a left mouse button click to advance and a right mouse button click to reverse.

Deleting polyspolygons

Select a poly that is not grouped, or use <Ctrl> to select only one. Click the Delete Poly gadget (LMB); Aladdin 4D asks for verification. Answer OK, and the polygon is deleted. It is truly gone. Once you have deleted a poly, you cannot get it back - so be sure when you answer OK in the requester that it really is OK to delete the poly.

Drawing splines

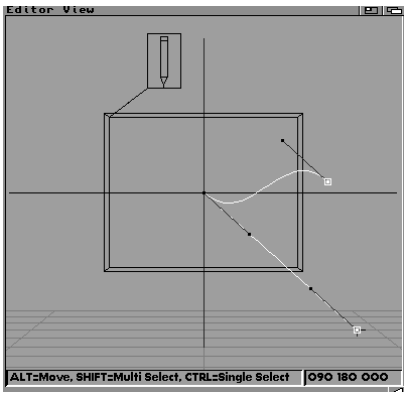
Now that you have a good basic understanding of drawing a freehand polygon, let's look at a second type of drawing that is considerably more powerful. Click the Freehand Spline gadget with the left mouse button. Move the mouse pointer into the Editor window, hold down the left mouse button, and drag. You'll see a line emerge from the current Attach Point (wherever it is) just as when drawing a polygon. There is a difference, however. The line has two red lines that change length as you move the active point. Use the <TAB> key to deposit the point and create another, and drag it away. Notice that you now have a triangle, with the red lines dividing each side into thirds. Invoke the Set command.

The triangle you have set is not a polygon - it's a spline. The red lines are the control vectors for the spline. The spline is a mathematical curve defined by its control vectors and anchor points. It's like a rubber band stretched between the two anchor points, with the two control vectors exerting pressure to make it curve, in proportion to the distance and directions they are from the anchor points. You can move all of these points around to control the shape and size of the resulting spline curve. The main points of the spline are at the points of the triangle. The control points are at the ends of the red lines emerging from each main point in the spline. Select a control point (one of the ends of the red lines) and move it, just as for a normal polygon. You'll see that the opposing control point adapts to the new angle and distance of the one you're moving, insuring that the curve is evenly modified.

You might also want to move a point independently. You can do this by holding down the <Ctrl> key while dragging the point. Holding the key allows you to move the main point without moving the control

points. The <Ctrl> key gives you control over the curvature of the spline at the main point it's connected to.

Set the spline. Select it by one of its main points and move the main point. If you hold down <Ctrl>, the main point and its two control points move at the same time. Of course, if you hold down the <Alt> key, the entire spline (and all other selected splines, and polyspolygons) will move.



You can add and delete points on a spline just as for a regular polygon. However, you must work with the main points, not the control points. After you add the points you want, then edit the control points to get the curves you want. You can also hide, show, and move splines just as with the regular polyspolygons.

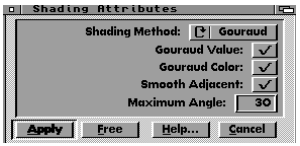
One aspect of the splines you may notice is the "smoothness" of the curves drawn in the Editor. When drawing splines, the Editor uses the size of each segment on screen to decide how smoothly it must draw the segment to be convincing. This technique considerably speeds up the drawing of splines. If you have the spline very small on screen, you may notice a multi-sided effect in a small segment. This is normal - it's just the screen representation of the segment. Within the program, it's still actually quite round. You can verify this by zooming in. As the segment gets larger on the screen, the Editor can resolve the spline in better detail for you to see. The Spline Editor does allow non-constant splines - X axisX-axis *OR or* Y axisY-axis different.

Splines are useful for creating more precise polygons that make up curved surfaces. The splines themselves do not render. They are always converted to polygons for rendering.

Conclusion

Now you have a basic understanding of the Editor's general operation. With a little practice, using it will become second nature to you. The polygons you create are the building blocks for many of the more advanced tools you'll want to use. While some tools, like the primitive generators create complete 3D objects, many others require a "template poly" to operate on. You now know how to draw these template polyspolygons. There are also built-in tools to generate arcs and rectangles. Other operations on these polygons will come later. These include snap and grid snap, extrusion, lathe, slant, stretch, clone, mirror, resize, etc. We'll touch on some of these as we go through the rest of the tutorials, but they're thoroughly described in the Reference Sections.

Tutorial: (Shading)



This section discusses the three types of shading available in Aladdin 4D: Facet, Gouraud, Maximum Angle, and Phong.

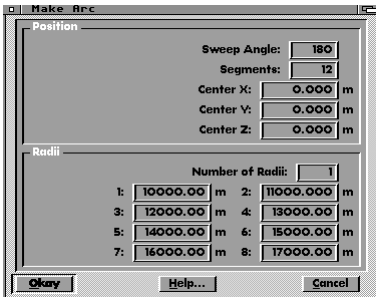
Facet shading (also called diffuse) is the fastest to render. It is the default shading for rendering the polygons. This type of shading uses one intensity and color for each polygon. It gives the appearance of each polygon in the object being a "facet" of a complex shape. If you don't set a shading type, or free shading, facet ~~is used.~~is used. Facet shading is quite useful for checking positioning, sizing, and other factors before devoting time to a full render.

Gouraud shading is the next fastest. This type of shading averages the intensity (value) and/or color from adjacent polygons to smooth the transitions from one polygon to another. It can blend colors of base polygons for effects that cannot be accomplished using Phong.

Another possible shading, introduced in more recent versions of Aladdin 4D, is the "Maximum Angle" type. This makes it possible to shade an entire object at one pass.

Phong shading is the slowest, though it delivers the truest lighting. Phong shading performs mathematically complex averaging of the normals of adjacent polygons to get a precise "picture" of the exact lighting angle, color bias, etc. It also maintains the exact 3D coordinate under examination, along with its distance and angle to each light. It has the ability to create highlights for rounded forms. It is also the only shading that will allow a gradient fall off across a single polygon. Many features of attribute and texture lists require Phong shading to be active.

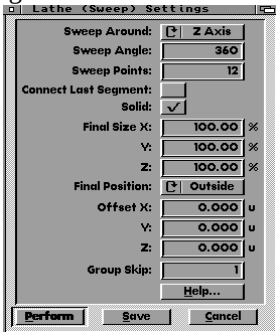
You can apply different types of shading within the same drawing to different objects, as your needs dictate.



To get started with this tutorial, we will need something to shade, so let's first make a simple sphere. You could use the Primitive tool to make this, but let's use an arc and the Lathe tool instead.

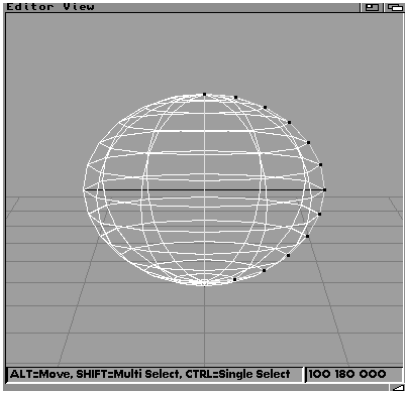
Flat View in the ~~Y axis~~Y-axis. Use the Make Arc tool to make an arc of 180 degrees and 12 segments.

Open the Lathe defaults (RMB on Lathe gadget or pull down the Edit / Tools / Lathe menu) and set Sweep Angle to 360 degrees, Connect Last Segment should be OFF (the Last Segment is the long straight line, and we do not want it to lathe. If it does, there will be a "stem" in the center of the sphere, causing strange shading patterns). Turn "Solid" ON. Set Sweep Around to the ~~Z axis~~Z-axis. This is the axis on which the Lathe operation will spin.



To operate all of the Aladdin 4D tools, you can establish their settings first, by accessing the appropriate requesters. These appear when you click the right button over the tool's name (most of them, anyway) in the External Tools box. You can always get to the tool settings by using the pull-down menus. If you select an object before setting up the tool to your liking, you can click the Perform button to have Aladdin 4D proceed with the action you want (where appropriate, of course). Some settings requesters also permit a disk save, from which you can later reload.

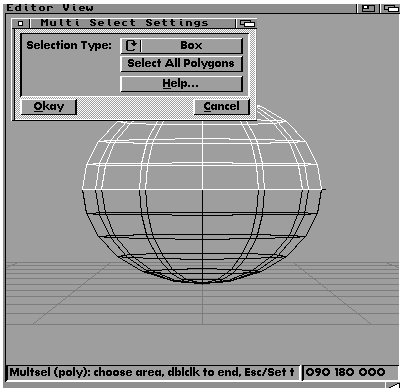
Accept the requester.-. Select the arc, and click the Lathe gadget.-. In a moment you'll see the sphere.-. Set the ~~polys~~polygons.-. Then select the template poly (the arc) and delete it. (Don't delete the sphere!) The template poly is actually hidden in the lathe operation, but it's easy to select since it was created first.



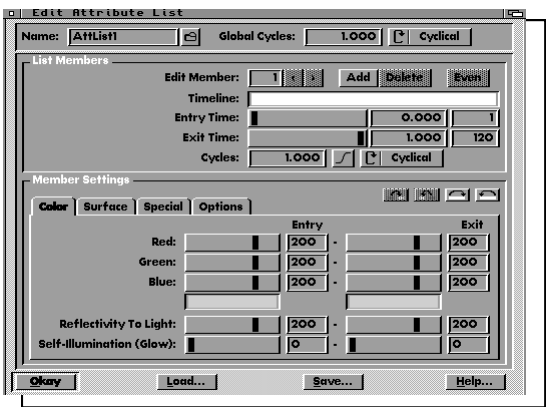
This is the object we will shade.-. Let's color the top of the sphere differently from the bottom.-. Go to isometric view mode.-. Use the multiple select tool to select only the top half of the sphere.

The multiple select tool itself has several available settings.-. You can view these by clicking the right button over the tool.-. For this operation, the "Box" setting is a good one.

Applying attributes



Tap the <a> key, or pull down the Object / Attributes menu.-. The Attribute List Selection requester opens.-. Since there aren't any attribute lists to select, we'll make one.-. Click New, which opens the Edit Attribute List requester.-. The first item in this requester is Name.-. This is where you can create a name for the new Attribute List.-. Name this one "blue".-. The Edit Attribute List requester is tabbed for organized presentation of its information.-. The first tab is Color, bearing sliders for each of Red, Green, and Blue components of the color you wish.-. If all of these are to the far left, the color is black.-. If all are maxed out to the far right, they'll read 255, instead of 0, and the color is purest white.-. These sliders operate on a 24-bit numbering scheme, regardless of the target picture you're creating.-. To translate 24-bit numeric settings for 16-color displays, divide by 8.-. If all three sliders are at the same numeric value, no matter what the value is, the color selected is a gray.



Blue color occurs when the Blue slider is farther over than the other two.-. The less of the other two colors, the purer the blue.-. Adding equal amounts of the other two colors to a maxed-out blue slider gives ever brightening blues, going pastel-ish as the other two sliders's matched values approach 220 or so.-. For this tutorial, set the Blue color slider to about 200 (its default, actually), and change the red and green sliders to 50 each.-. There are two sets of these sliders, one each for Entry and Exit values.-. The Entry sliders define the color applied to polygons at the beginning of an animation, and the Exit ones apply to the color at the end of the sequence.-. During the animation itself, Aladdin 4D will calculate a position between the Entry and Exit settings that is proportional to the current frame's position in the animation, beginning to end - Entry to Exit.-. If you're creating a single picture, rather than animation, you can ignore the Exit values.-. Make them both the same, this time.

Match gadgets

Did you type the Blue value into the little box to match Exit to Entry? ? There's a cool way to do that. . At the top of the Member Settings panel of this Edit Attribute List requester, there are four little graphic gadgets. . The two gadgets at the far right let you easily match Entry and Exit settings in the gadgets below. . The rightmost one matches Exit to Entry gadgets, and the one next to it matches Entry specifications to Exit ones. . If you want matched entries, you can set up either one of the columns of entries to your liking, and then click the appropriate Match Gadget to copy them to the other column. . These gadgets occur throughout Aladdin 4D's requesters, and though this manual doesn't always mention them, they're there for your use. (If they don't apply, they're ghosted).

Match gadgets

Did you type the Blue value into the little box to match Exit to Entry? ? There's a cool way to do that. . At the top of the Member Settings panel of this Edit Attribute List requester, there are four little graphic gadgets. . The two gadgets at the far right let you easily match Entry and Exit settings in the gadgets below. . The rightmost match gadgetone copies thematches Exit values to the Entry gadgets, and the match gadgetone next to it copiesm theatches Entry valuesspecifications to Exit gadgetsones. If you want matched entries, you can set up either one of the columns of entries to your liking, and then click the appropriate Match Gadget to copy them to the other column. . These gadgets occur throughout Aladdin 4D's requesters, and though this manual doesn't always mention them, they're there for your use.

Here's what the Match Gadgets do:

Match Entry: The exit values will be changed to match the entry

Match Exit: The entry values will be changed to match the exit

Match Last Exit: The entry values will be changed to match the last member's exit values for smooth member transition.

Match Next Entry: The exit values will be changed to match the next member's entry values for smooth member transition.

Match Entry: The exit values will be changed to match the entry.

Match Exit: The entry values will be changed to match the exit.

Any of these that don't apply to the present Attribute List being edited will be ghosted.

Okay, back to the tutorial... The Attribute List named Blue has matched blue settings. . Set Reflectivity To Light to 255 (instead of 200). . Leave everything else as it is, and move to the Surface tab.

Set Hardness (Specular) to 255, and Highlight Size (Gloss) to about 200 (the default). . Do the same changes for both Entry and Exit (Remember those Match Gadgets?), and Accept the requester(s).

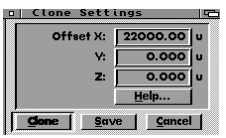
In Aladdin 4D, applying attributes is the way you get lots of things to happen that you can use and re-use as you work.. Notice that the requester is a two-step operation.. You can select lists of available attributes from the first requester, without going into the one you get if you select "new." The ones in the first list are the currently available ones.. There were none listed because none had been created. Now that you have an attribute list named "blue" in place, it will be listed in among the Texture Lists, and you can select it without dealing with the complexity of the second requester unless you need to. You can save more lists based on "blue" or anything else, and apply them to any objects you choose.. Aladdin 4D leaves the naming scheme up to you, but you should use a system that's easy to remember - like naming the attribute sets like the objects to which they're applied, "tires" for example, or "glass-globe."

Set the polyspolygons, and use the Multiple Select tool to select the bottom half of the sphere this time.. Tap the <a> key again - note the presence of "blue" - and click New.. Name this new list "yellow", and set its RGB sliders to 200, 200, 0, respectively for both entry and exit.. Set everything else as before Reflectivity To Light (in the Color tab) 255, Hardness (in the Surface Tab) 255, Highlight Size 200.. Accept the requesters, and the bottom half of the sphere will now have these attributes.. Set the polyspolygons.

Just for good measure, let's do one more.. Use multiple select (<Shift> or use the toolbox's convenient Multi tool) to select the middle two rows of polygons.. This time when you bring up the Attribute Lists Selection requester (the <a> key) you get a requester telling you that the selected polyspolygons use different lists.. OK this requester, and the "blue" list is selected in the window.. Click New, and name this new list "red". Give it Red, Green, and Blue values of 200, 0, 0 on both Entry and Exit.. Keep everything else the same (Reflectivity, etc.) as the "blue" list.. Accept the requesters, and the middle section of the sphere will display these attributes when rendered.

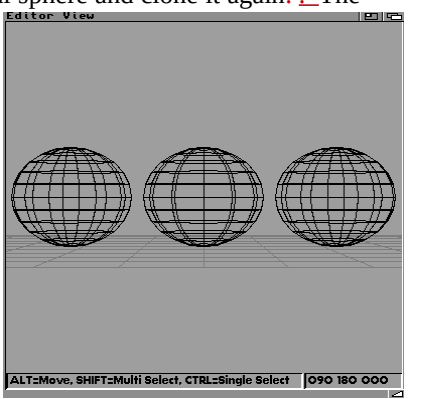
A quick review: The top portion has the attributes defined as "blue." The middle has "red", and the bottom of the sphere has "yellow."

We want to show off each type of shading in this same drawing, so let's clone the sphere.. Open the Clone defaults by clicking the right mouse button on the Clone gadget (or pull down the Edit / Tools / Clone menu or use the <Amiga> <c> keyboard shortcut).. The Clone Settings requester contains three numeric boxes for Offset X, Y, and Z.. Set the X offset to 22000; leave the others at 0, and accept the requester.



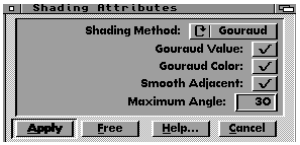
If you select the sphere before opening the requester with the pull-down menu, you can click Clone to do the deed.. Once you set the requester up, however, it will remember its settings for further use, so next time you want to clone something with 22000 X offset, you can simple click the tool gadget (LMB) or use the keyboard shortcut.. Simple!

Whichever method you use, clone the sphere.. Set the new objects.. Open the clone defaults again and set the X offset to -22000.. Accept the requester.. Select the original sphere and clone it again.. The result is three spheres lined up across the screen, all of them facet shaded (the default), and all of them having blue tops, red middles, and yellow bottoms.. Wonderful.. Set the spheres.. Click the isometric gadget to go back to perspective view.. Tap <Cursor Up> a few times so you can get a good look at this lineup of spheres.



Now to change the shading.: Select the sphere in the middle.. Select the Object menu item Shading.. The requester that opens allows you to specify which type of shading you want applied to the object you've selected.. The available types of shading are

selectable with the ~~rotary gadget~~cycle gadget at the top of the requester~~.~~. The available choices are None, Gouraud, and Phong~~.~~. The other check boxes specify Gouraud Value, Gouraud Color, Smooth Adjacent and Maximum Angle~~.~~. Turn on Gouraud Value and Gouraud Color, but leave Phong OFF, and Smooth Adjacent ON.



Accept the requester~~.~~. This time you see a progress meter appear and fill up as Aladdin 4D pairs adjacent polygons. (Common nodes in the polygon net are being recorded.) This can take some time for a large number of polygons~~.~~. Aladdin 4D does this step here, instead of having to do it for every frame in an animation~~.~~. When finished, the racetrack window will close, and the shading is applied as you established~~.~~. The center sphere now has Gouraud shading~~.~~. Select the right sphere~~.~~. Get the shading requester again, and turn ON Phong shading~~.~~. Again leave the Smooth Adjacent ON, and accept~~.~~. The right sphere now has Phong shading.

Note: Smooth Adjacent was called Assign Adjacent in previous versions of Aladdin 4D.

The three spheres are ready to render~~.~~. Press <Shift> and <F9>~~.~~. Accept the requester~~.~~. The Editor Screen closes and the render screen opens~~.~~. After the initialization you see the spheres painting They are in wireframe because this is the display mode selected~~.~~. You could have selected the mode from the Editor, but it is good to see what this looks like~~.~~. Choose the Render menu item Display Mode~~.~~. Turn ON Light, Fill, Gouraud, and Phong~~.~~. Accept the requester~~.~~. If you like, and have the extra hardware, change the display type to use your 24-bit display board (or DCTV)~~.~~. This will force a redraw~~.~~. If not, select the Project menu item Redraw or just tap <F9>~~.~~. This rendering will illustrate the differences among the three shading types.

Facet shading uses one intensity and color for each polygon~~.~~. Gouraud shading is smoothed and shows off its ability to blend colors of base ~~polys~~polygons~~.~~. The Phong shaded object looks quite smooth, and because of its ability to have highlights, looks hard and glossy~~.~~. The colors of the base ~~polys~~polygons are not blended in Phong.

Maximum Angle Shading

In the Shade requester, a gadget called "Maximum Angle" makes it possible to shade an entire object at one pass~~.~~. You simply enter the Maximum Angle that you want for Smooth Adjacents~~.~~. The angle is ignored if Smooth Adjacents is off~~.~~. You can select and shade only the ~~polys~~polygons you want~~.~~. When you use Maximum Angle Shading, instead of the more calculation-intensive options, the speed of the shading routines dramatically increases~~.~~. Of course, you can still shade manually for full control when needed.

The Shade tool allows you to free ~~polys~~polygons that have applied shading and have duplicate points~~.~~. This condition can arise if you shade a group of ~~polys~~polygons and then resize them so that points overlap~~.~~. Use the "Free" button in the Shading Requester to remove shading from any ~~polys~~polygons or objects you have selected~~.~~. You could also set the objects' shading type to "None" using the ~~rotary gadget~~cycle gadget.

Smooth Adjacent is set by the ~~polys~~polygons you select to shade~~.~~. If you have previously shaded the ~~polys~~polygons with this flag on and at least two share a point in space, the Smooth Adjacent option will be set to on~~.~~. If you have selected previously unshaded ~~polys~~polygons, or if they were not shaded previously with this flag on, the requester will open with the option off.

Also if you select polygons out of two or more previous shading groups to shade, you'll be asked to confirm that you wish to change the current shading groups before shading takes place.

Palette

~~/// WHERE IS THE PALETTE OPTION described here ??? 0415 /J~~
~~/// ??? 0504 /j~~
~~@ If you're rendering in a HAM or Ham-8 mode, the display you just saw~~
~~@ isn't very satisfying. This is~~
~~@ because the program's palette defaults to all greys for its Ham modes. An all~~
~~@ grey palette is a good~~
~~@ general choice for busy images, but for this one, which has lots of pure~~
~~@ colors, it can be much better. Use the Render/ Render Settings menu item.~~
~~@ The Color tab controls the palette of the rendered picture,~~
~~@ when in HAM modes, establishes the leftmost 16 colors for the program to~~
~~@ use to generate the other colors it needs to create your picture.~~

~~/// THIS DOES NOT APPEAR TO BE THE SAME THING... delete it ??? /j~~
~~@ allows you to choose a color square, then pick the color~~
~~@ for that square directly from the screen. Choose several representative~~
~~@ colors from the picture for the palette. Choose those that occur most often,~~
~~@ or those that have the worst fringing. Now open the Display Mode requester~~
~~@ again, and turn ON the Palette Match. Redraw the image and you should see~~
~~@ considerable improvement.~~

~~@ The "bands" of color occur because of the~~
~~@ limited number of colors available in HAM mode. These bands are called "mach~~
~~@ banding". To reduce this effect, turn on Dither (in the Display Mode~~
~~@ requester) and redraw. Dithering is applied after rendering, as a smoothing~~
~~@ factor. The introduction of some patterning of noise, or dither,~~
~~@ will reduce or eliminate mach banding. However, the best defense~~
~~@ against mach banding is to use a very busy image, of the type you get using~~
~~@ a digitizer, and putting a texture on the polygons. The resulting~~
~~@ image is far superior to the one you see - as you'll find out in the next~~
~~@ section. If you are using a 24-bit board you probably wonder~~
~~@ what we are talking about here, as you don't see any banding. If you~~
~~@ really want to look at the problem, render a HAM image just to see.~~
~~@ If you are using DCTV, you may experience some NTSC-type bleed between~~
~~@ colors that are too pure. DCTV displays an NTSC image, and NTSC is~~
~~@ limited in its ability to transfer pure colors, as well as in its ability~~
~~@ to resolve strongly contrasty transitions. Textures, (again, a busy image)~~
~~@ will also help this problem by diffusing the problem over many pixels.~~

Small Palette Picture Quality

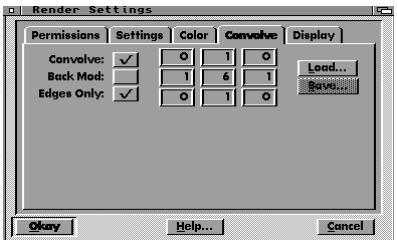
In a display of limited palette size, the most difficult render is exactly the type of smooth color transitions that you see here.

To reduce banding and control other types of small-palette troubles, you can resort to external programs to operate on the rendered images. For example, you can also save the images as 24-bit Amiga IFF format pictures, and then use a paint program or image processing program such as ImageFX to load the 24-bit picture and convert it to HAM for viewing. You can even use Aladdin 4D's Frame Script feature to automate this process somewhat.

The resulting pictures will be superior to what you can get in Aladdin 4D, because the work you do on the image will occur after the renderer has done all of its work. When the image exists as a whole, image processing techniques can make better decisions for palette, and perform a picture-wise dither to simulate a much larger color palette. As the image renders in Aladdin 4D, such decisions aren't possible, since it is unknown what colors will exist in the final image until it's done - and then it's too late. These choices aren't necessary in 24-bit rendering, since all the colors needed are always present.

Limited-palette problems disappear when you switch to a 24-bit render. For practical purposes, a 24-bit picture contains the entire palette of visible colors, so there's no need to worry about any limitations. For drive space and processing time, however, the Ham-8 format gives excellent latitude and resolution, and usually is capable of delivering 24-bit quality pictures without the 24-bit overhead.

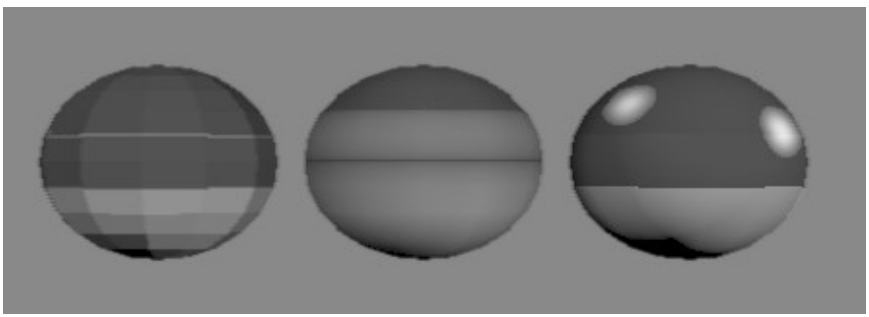
Convolve



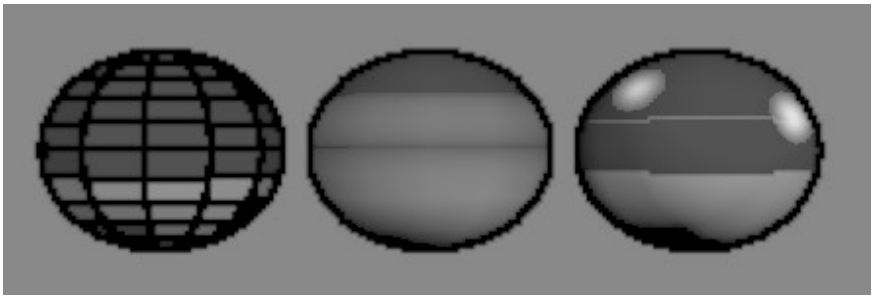
This is a good place to take a look at Convolve. The center and right spheres do not show any "jaggies" except at their edges, where the background meets the edge of the spheres. The faceted sphere has jaggies all over. Aladdin 4D has an efficient way of concentrating only on the problem areas. The program can apply an inline Convolve, talked about more extensively in the reference sections. Use the Render Settings requester tab Convolve.

Edge Only: The Aladdin 4D Convolve is able to do edge detection - not just a color threshold. It can tell if polygons participate in the same shared list at edges, and it can detect when polygon/background edges have been hit. If you have Edge Only selected, the convolve will occur at the edges of ~~polys~~polygons and backgrounds, and ~~polys~~polygons that are not shaded together only.

Turn ON Convolve and Edge Only, and accept. Re-render the drawing and you'll see that the areas of the drawing that have jaggies are the areas that are smoothed. The special thing about this is that it is ONLY those areas. To prove this, enter zeroes in all nine Convolve gadgets. This is not a legal convolve, and Aladdin 4D will recognize this and print black where the convolve would occur. Re-render and you'll see clearly just where the Convolve is. This odd effect is only for discussion. This type of black convolve is not meant to be a part of any display. There are several types of pre-defined Convolutions on the disk for you to load and use, and you can define and save your own, as well



You can also choose to convolve entire objects, and the background, or you can omit the background from the operation.



Antialiasing

Although Convolve does a fine job of smoothing for some drawings, and is very fast, there are times and drawings that call for something more~~.~~. Antialiasing does that~~.~~. It basically performs a smoothing operation to reduce the jaggie look - especially useful when objects contrast drastically with adjacent colors like backgrounds or other objects. Antialiasing in Aladdin 4D is selectable as low, medium, or high~~.~~. Low antialiasing can be applied horizontally or vertically~~.~~. Medium and High modes always affect both directions regardless of the settings.

Open Render Settings and turn OFF Convolve~~.~~. The Settings tab gadgets control "super-sampling", an antialias function~~.~~. When these are set to other than "none", Aladdin 4D creates a much larger "sample" than necessary, averaging the results to obtain the final specification for a given pixel~~.~~. The operation can be performed horizontally or vertically - or both.

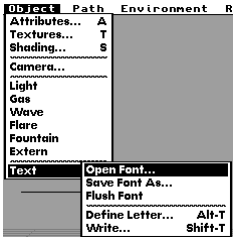
Super-sampling can have four settings in each of vertical and horizontal directions.

Horiz~~.~~. Super-Sampling (~~rotary gadget~~cycle gadget): This gadget can read None, Low, Medium, or High~~.~~. It's the control for antialiasing (global) in the horizontal direction.

Vert~~.~~. Super-Sampling (~~rotary gadget~~cycle gadget): This gadget, also, can read None, Low, Medium, or High~~.~~. It's the control for antialiasing (global) in the vertical direction~~.~~. The settings need not match~~.~~. That means you can set as much super-sampling as you need, to get rid of artifacts in that direction, without putting up with the extended render times of super-sampling in a direction you don't need~~.~~. Of course, if you need both, turn them both on.

The "High" antialias setting~~s~~ render~~s~~ 16 rays per pixel~~.~~. The "Medium" antialias setting renders 9 rays per pixel. The “Low” setting renders 4 rays per pixel.

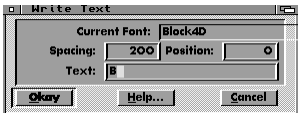
Shading groups



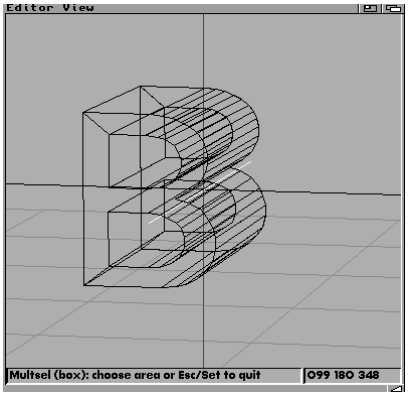
Now that you have a more satisfying image to work on, lets get back to shading, which is, after all, the subject of this section~~.~~. Return to the Editor~~.~~. Select all of the ~~polys~~polygons in the drawing (Use the Multiple Select gadget) and delete them~~.~~. This leaves your Attribute Lists in place, but no objects~~.~~. If you selected "New", the Attribute Lists would also be history~~.~~. Now go to the Object menu

and, under TEXT, choose Open Font~~.~~. Go to your Aladdin 4Dfonts drawer and open "block.4dff." Aladdin 4D's fonts resource can be assigned anyplace you like on program startup~~.~~. See the section on Tooltypes.

After a few seconds, the font will be loaded~~.~~. Go to the Object menu item "Text" and choose "Write." The Write requester contains some other numeric entry boxes, but leave them at their defaults for now.



At the bottom of the requester is a string entry gadget~~.~~. This is where you provide the text you want Aladdin 4D to convert to objects for you~~.~~. Enter just a single letter, "B" (capital B) and accept~~.~~. You'll see the capital "B" appear at the origin~~.~~. Select this object and click scale it using the Scale tool~~.~~. Scale is accessible from the External Tools box, or from the Edit / Tools / Scale pull-down menu~~.~~. The keyboard shortcut is <Amiga> <L>~~.~~. You can also simply specify a size in the box that makes the object in the first place..~~.~~. but that's too easy, and wouldn't lead us to explore the Scale tool, anyway.

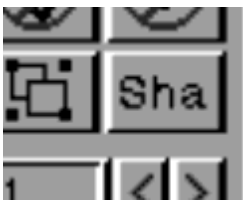


In the Scale Settings requester, enter 400.0, 400.0, 400.0 in the Scale XYZ gadgets and click Scale~~.~~. We want to scale from Center (The ~~rotary gadget~~cycle gadget in the middle of the requester), and we don't want to deform on the fly, this time, so leave the other gadgets at their defaults~~.~~. The Scale_a gadget is interactive, so let it be for now~~.~~. Just use the Scale Settings requester to enlarge the object 400 percent in all directions~~.~~. Even so, it's still flat~~.~~. So...

Set the poly~~.~~. Immediately again select it and Extrude it~~.~~. The Extrude settings requester is accessible from Edit / Tools / Extrude, or by clicking the right mouse button over the Extrude gadget~~.~~. Use 10000 units on the Y axis, 1 segment~~.~~. Click Perform to start the extrude~~.~~. You now have the letter B extruded way into 3D space~~.~~. Way cool~~.~~. Set the poly.

Now let's consider how this letter should be shaded~~.~~. First, when you shade some polygons, the program figures out whether any two (or more) polygons have a common point in space~~.~~. If so, it records this information and shades these together when rendering~~.~~. ~~Polys~~Polygons with more than four points are never shaded together~~.~~. So if you simply select all the polygons and shade them, the program will shade across the corners of the "B" object at the top right and bottom right~~.~~. There are some other places that we don't want shaded~~.~~. So we'll tell the program this by shading different sets of ~~polys~~polygons which will become the shading groups.

Remember, we're talking about shading, here, not shadows~~.~~. Shading is the scheme the program uses to make solid-looking planes defined by the wireframe you see in the editor.



To shade in different sets, simply select those ~~polys~~polygons that you want in a shading group and shade them with no other ~~polys~~polygons selected~~.~~. Keep in mind that if two (or more) ~~polys~~polygons do not share a point in space, it makes no difference whether they are in the same shading group, since they will not be shaded together, regardless~~.~~. So for the extruded shell of this letterform, the shading sets could be like this (We have, of course, separated them for clarity): Set 1 is shown on the left, set 2 in the center and set 3 on the right~~.~~. There are other possible combinations, but as you can see the polygons that share points at the junctions that we don't want to shade across are in separate sets, as desired~~.~~. The technique for selecting the ~~polys~~polygons

~~for a Shading Group is quite simple,~~polygons for a Shading Group is quite simple and after a little practice comes very naturally~~.~~. Lets do this for the letterform now.

We'll use one of the group levels as a temporary group~~.~~. You are currently in Group Level 1, so let's use it.

First select the front face of the extrusion~~.~~. As you can see, extrusions are always grouped with the shell separate from the faces so only the faces select~~.~~. Hide them~~.~~. Now you have only the shell visible~~.~~. The shell is grouped, so if you select any poly of the shell, the whole shell selects~~.~~. Hold down <Ctrl> and <Shift>~~.~~. The <Ctrl> key says "select only one poly out of a group," and the <Shift> key says "allow another poly to be selected"~~.~~. Now place the mouse pointer over one of the ~~polys~~polygons that make up what we showed as the third set~~.~~. Hold down the left mouse button and move it over these ~~polys~~polygons~~.~~. You will see them selecting~~.~~. Keep going until you have all of them selected~~.~~. If you accidentally select an adjoining poly you didn't want, set the ~~polys~~polygons and start again~~.~~. Once you have them selected, click the left mouse button on the Group gadget to group them and you'll see them set~~.~~. Now select them and hide them~~.~~. Use the same procedure to select all the ~~polys~~polygons in what we called the second set and group them~~.~~. You don't have to do the others, since they are still grouped by the extrusion process.

The shell is now grouped (in Group Level 1) the way we want the shading groups~~.~~. Show all the ~~polys~~polygons~~.~~. Select the faces of the letterform~~.~~. Choose the Object menu item "Shading" and turn on Phong shading for them (It doesn't matter for these whether Adjacent is on)~~.~~. Accept the requester and hide them~~.~~. Now select one of the other groups, shade them with Phong, and turn Smooth Adjacents ON, then hide them~~.~~. Do the same for the other two sets of ~~polys~~polygons~~.~~. As a final step, show all the ~~polys~~polygons, select them all, and group them in Group Level 1~~.~~. This whole process takes far longer to read than it does to do~~.~~. Work through it carefully, though, to make sure you understand this important process.

When you shade ~~polys~~polygons, the program automatically groups those shaded at the same time into a Shading Group~~.~~. Move the mouse pointer over the area that displays the Group Level and click with the right mouse button until you see "Sha" displayed~~.~~. This is the Shading Group~~.~~. You cannot group or ungroup while this is displayed, but you can select ~~polys~~polygons and see what ~~polys~~polygons are in the same Shading Group~~.~~. For any object that was not shaded, each polygon will select independently of the others~~.~~. If you have shaded the object, selecting one poly will select all others that were shaded at the same time~~.~~. This provides a convenient method to re-shade the ~~polys~~polygons, if desired, and to see what shading groups were used if you get a questionable render~~.~~. It also allows you to study the shading groups when you load a drawing created by someone else.

Animation Tutorial (Animation)

This section contains tutorials covering polygon animation in Aladdin 4D, which contains one of the most flexible and comprehensive animation interfaces in any 3D program.

There are really several types of animation in Aladdin 4D. There are Texture and Attribute lists, lens flares, gas turbulence, camera translation/tilt, target tracking, background/foreground/overlay compositing, wave sources, etc., etc. There is also the type of animation where polygons deform, move along paths and rotate, scale, create mechanical waves and instance. To make things even more fun, all of these things can happen at the same time! All give you great freedom to do almost anything you want. The key is to learn each type, one at a time, and then combine them to achieve the effects you want.

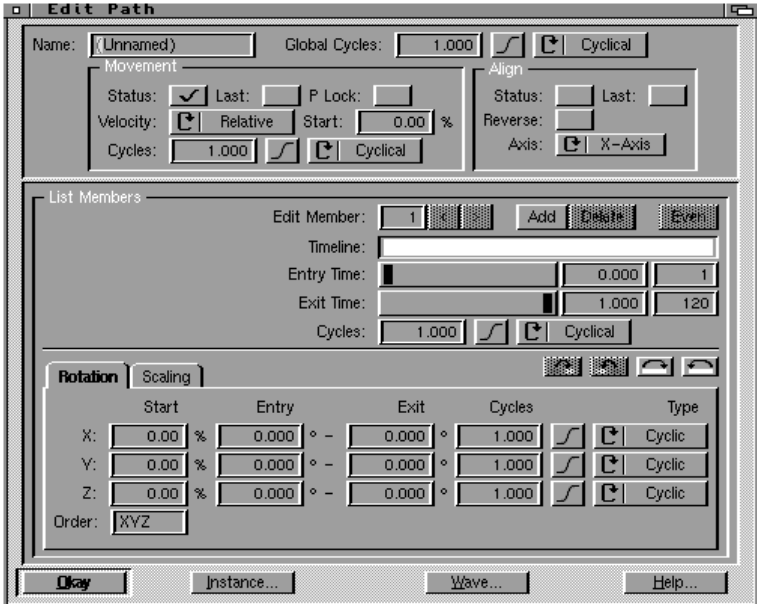
Here, we'll concentrate on the polygon animation types. They are translation, rotation, scaling, deformation, mechanical waves and instancing. We will also take a look at camera-target animation. For this tutorial, it is assumed that you have read the quick start and worked through the tutorials, and of course have mastered operating the Amiga.

Translation

Translation, or moving polygons is the easiest type of animation. Select "New" and Flat View in the Y axis. Use the Object menu item Make Arc. Enter 360 degrees and 3 segments. Accept the requester. Select the resulting triangle. Click the Scale_a gadget with <Shift> and the left mouse button (to resize in all dimensions at once); resize it to about 20 percent of the original. (You can also just enter the Scale Settings requester, enter 20% for X, Y, and Z, and click Perform.) Set the triangle. This is the polygon that we are going to move around.

Set the Attach Point to the origin (RMB on Page Move gadget with no polygons selected). Click the FreeHand Poly tool and draw a line to the right, almost to the edge of the screen. Set it. This straight line is the polygon we will use for a path.

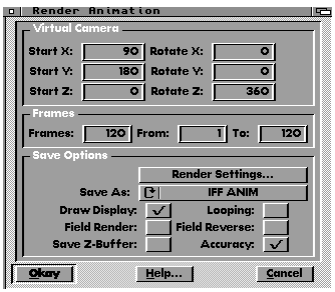
Select the straight line, and select the Path menu item Make Path. The Path requester opens. Notice that you can give a path a name - very useful for keeping track of complex motions of lots of objects on the screen. The Path requester is divided into logical areas, with tabs at the bottom. Movement is defined in the top section - here, you should turn Status ON, to permit translating the objects down the path. The Path requester does many tricks in animation, so it has lots of other parameters. Alignment is defined in the smaller box to its right, and time line specifiers are entered in the center panel. The bottom, tabbed panel gets down to the nitty-gritty of rotation and scaling.



We only want the triangle to move along the straight line, so just leave everything else alone and accept the requester.~~-~~. The line is now a path, and it's red, to distinguish it from part of an object.~~-~~. This is the color of paths in the Editor.~~-~~. Currently, however, it's only a path.~~-~~. In order for any action to take place, something has to follow the path, and we have to tell the program what objects we want to apply.~~-~~. Select the triangle and then select the Path menu item Assign Path.~~-~~. The program prompts you to choose a path.~~-~~. Select the straight line.

You can have the program show you what ~~polys~~polygons are assigned to paths (and vice versa).~~-~~. The Path menu items Show Assign Polygon and Show Assigned Path do this.~~-~~. Select an object, and pull down Show Assigned Path, and the program will highlight the path to which it's assigned.~~-~~. Show Assigned Polygon works the same way - but select a path first.

Render the Anim



We are ready to look at the motion.~~-~~. Select Render / Preview Anim.~~-~~. The requester lets you define a global rotation (if not using a camera) and the number of frames to animate, as well as how to save - as Amiga anim or single frames, etc.~~-~~. Change the number of frames to 120 and the From and To gadgets to 1 and 120, respectively.~~-~~. The Preview screen opens, showing you the animation as specified.~~-~~. While in the preview screen you can change view direction and zoom factor in the same manner as in the Editor.~~-~~. Try this.~~-~~. It's detailed in the Reference section on previewing animation.

~~#### screen illus of tutorial in progress /j~~

~~**HARVNOTE - SUBSTITUTED SCREENSHOTS ABOVE. . THE WIREFRAME PREVIEW ANIM IS NEARLY TOTALLY BLACK AND WILL NOT PRINT WELL AT ALL.**~~

Edit Path

Press <Esc> to return to the Editor.~~-~~. Select the path and then the Path menu item Edit Path (formerly known as "Alter A Path").~~-~~. In the "Movement" panel, next to the Status gadget is a check box gadget called Last.~~-~~. Turn this ON and accept the requester.~~-~~. Press <F9> to preview.~~-~~. To Aladdin 4D, a straight line is a polygon, and actually has sides even though they are right on top of each other.~~-~~. Think of it as a triangle where one of the sides is infinitely small.~~-~~. The triangle object moves along the first long side, across the infinitely small side - which is imperceptible - and then along the last long side.

Return to the Editor. Select the path by its right end and tap the <TAB> key. You have entered Freehand mode. A new point was created when you hit <TAB>. Move it down the screen some place. Tap the <TAB> key again, and move the next new point. Do this several more times. Press <F9> to preview the path's changes reflected in the movement of the poly.

Being a polygon of however many sides, the path has a first point (point 0). During the animation the first point moves along the polygon (unseen by you), and the change in XYZ position is measured. This same change is added to any polygons that are assigned to the path. This means that the movement is independent of the polygon's relationship to the path. To prove this, select the triangle and then click the Clone gadget with the right mouse button. Set X offset to 5000 and accept the requester. With the triangle selected, click Clone twice with the left mouse button. There are now 3 triangles. As the triangles were cloned they were automatically assigned to the path, since the first one was. Use Preview again to see that the movement is identical for the new triangles, even though they are in a different relationship to the path.

Return to the Editor and Edit Path again. In the Align section of the requester, the Reverse gadget reverses the direction of movement. The Plock gadget limits the movement to the points that exist on the path.

And the ~~rotary gadget~~cycle gadget in the topmost panel determines the type of motion: Periodic tells the program to move the ~~polys~~polygons along the path from start to finish; Cyclical tells it to move them from start to finish and back again. Global Cycles is the number of times you wish the activity to occur. If you want three loops, for example, enter 3. This item is different from the Cycles specification in the bottom panel, which applies only to translation.

Try these gadgets in different combinations, using Preview to observe the changes.

Other uses for Paths

Scaling ~~(~~Tab~~)~~

In addition to simple motion, Paths can control a number of other factors during animation. These are "translations" of the objects' shapes over time. An array of gadgets that relate to translation live in the Rotation and Scaling tabbed sections at the bottom of the requester you've been using.

The Scaling tab section of the requester contains Start, Entry, Exit, and Cycles numeric entry boxes, plus a funny squiggle - that's the gadget for the CSpline control function - and a Type gadget which rotates among Cyclic and Periodic, for each of X, Y, and Z axes. Most of these are fairly intuitive as to what they do, but here goes...

Start (X, Y, and Z) is the place you want the translation scaling to be (expressed as a percent of its normal size) when the animation begins (it will loop back to this as though it is also the end percent).

Entry percent is the Entry specifier, of course - again for each of X, Y, and Z. Exit.. ditto, but for the exit percent. With entry at 0 and exit at 100, the object would "grow" to its full size, for example.

Cycles is the number of times you wish the translation to occur.

This scaling can be under the control of one of Aladdin 4D's powerful CSplines. Clicking the Spline gadget (the squiggle) invokes Spline control for the item, and offers the Spline Editor to set it all up with. The CSpline gadget controls the Entry/Exit times and can implement ease-in and ease-out for natural looking motion and translation, all at the same

time. We'll come back to the massively powerful CSpline editor later. For now, just note its looks and presence. Many things in Aladdin 4D can be controlled with CSplines, most of them by clicking a gadget that looks like this.

In the Scaling tab, the ~~rotary gadget~~cycle gadget past the Spline icon tells the program whether you want the translation applied in Cyclic or Periodic manner. Periodic is the number of times through that you've established in the Cycles column. The four graphic buttons at the top of the tabbed panel specify the type of looping to be used.

Align

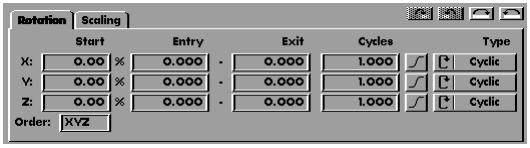
The Align option is controlled in the Align section of the requester at the top right. (If Status is on), Align tells the program to measure the angle of the current segment with the axis indicated and rotate the polygons to match. The current segment is actually averaged so that the rotation is smoothed between it and the previous segment. The axis you should use is always the one that is most perpendicular to the polygon. For this one, it is the Y axis. Turn ON the option, and select the Y axis with the rotary Axis selector. Preview the anim. The normal use of this is for larger paths that are smoother in their changes, but even for this jagged path, the averaging and rotation should work well. There are, of course, two ways the rotation could occur. If the rotation is opposite to what you want, just click the Reverse gadget in the Align area.

Animation Speed

The speed of the polygon along the path is controlled by several things. First is the number of frames in the animation. If the poly is moving along the whole path - you can limit the range of the poly movement with a CSpline - it will appear to move much more rapidly over 10 frames than a thousand frames.

Second, you can change the rate of speed at will with a CSpline. Third, you can have Aladdin 4D average the lengths of the path's segments in one of two distinct ways. The Velocity selector lets you choose Relative or Constant velocity. If Relative, the program will give each segment of the path an equal amount of the path's total time. This means that the polygons will move faster for longer segments, slower for shorter segments. This allows you to graphically apply speed changes to the ~~polys~~polygons and judge the result visually in the Editor. It's useful to simulate bouncing, for example, in an intuitive way. If the gadget is set to "Constant," the movement is divided along the entire length of the path. The ~~polys~~polygons will travel at the same rate of speed regardless of the length of the segments.

Rotation



Rotating the polygons requires a little more thought and understanding that scaling, so it's been left for last, though its tab is first. Edit the path and turn Align OFF. Leave Movement Status ON, Last Segment ON, and use "1.000" cycle with no CSpline. Accept the requester and hit <F9> to make sure the ~~polys~~polygons are only moving along the path.

Now edit the path again. Look at the middle part of the requester. The usual time line controls are there. The time line controls for the path do not apply to the translation. It is based on the shape of the poly. The time line does control when rotation (and scaling) occur. This segment of time is then further controlled by the independent CSplines available for these functions. You can always use a full span in the time line, then use CSplines for full control. The time line often simplifies the task, however.

In the rotation tab, rotation is specified as angles around the X, Y, and Z axes. "Entry" is the starting angle. "Exit" is the ending angle. If you have an Entry of 0 and an Exit of 360, the assigned polygons (and any hierarchy from this point down) will begin at 0 degrees rotation, and end rotated 360 degrees during the member's active time. If you have Entry set to 0 and Exit at -360, the same motion occurs, but in the opposite direction. If you have an Entry of 180 and an Exit of -360, the polyspolygons will rotate a total of 540 degrees, starting at 180 and rotating backwards. This scheme allows you to add a member to the rotation list for a path and insure that the starting angle is the same as the ending angle of the previous member for a smooth transition.

Enter an Entry of 0 and an Exit value of 360 in the Y. Accept the requester and preview the anim. The polyspolygons are indeed moving along the path and rotating once around the Y axis as specified. Important! Notice that the center of rotation is the point that is translating.

This is, as you found out in the translation tutorial, the first point (point 0) of the path. You can find out which is the first point by opening the manual mode. It's near the bottom of the Edit menu, or click the right mouse button over the freehand gadget. While the manual mode requester is on screen, clicking the points in the path will cause their numbers and coordinates to appear in the Manual Mode requester - remember you must set the object to change the point you're clicking. A polygon's point 0 is its first point. You can also choose a new first point, if desired. using the Edit menu item Choose First Point. For easiest use, click the polygon you wish to change by the point you want as its first point. Then invoke Choose First Point, and click the highlighted point again.

If you wanted the polygons to rotate around their center, you can simply snap them to the first point of the path. Let's do this. Select (or change) the first point in the path and Set the polyspolygons. This moves the Attach Point to that point. Now select the three polygons (<Shift> click) and press the right mouse button over the Page Center gadget. This centers the polyspolygons to the current Attach Point. Hit <F9> to see the result. The polygons now rotate around their common center.

The center of rotation is important. You should experiment with it until you understand what's going to happen. When you move a path, don't forget that the center of rotation also moves. Assigned polygons that rotate will change how they rotate unless they also move with the path.

The other controls for rotation are detailed in the reference section for paths.

Notice that each angle has a CSpline, and the member has a CSpline, and the globals area has a CSpline. As you may expect, these qualify each other in a natural division of time. In other words, if the CSpline you use for the global CSpline compresses time in the first half of the animation, the member CSpline(s) will react to this compression. This makes any type of rotation possible. Remember as you experiment, you need enough frames to see the rotation, especially if you compress it by a large amount.

If you use rotation along more than one axis in a member, you can pick the order in which the rotations occur. Often it is helpful to actually hold some arbitrary object up in front of you (a floppy disk or a pencil will do nicely) and go through the rotations in different orders to see the outcome. Then specify the order that produces the desired result.

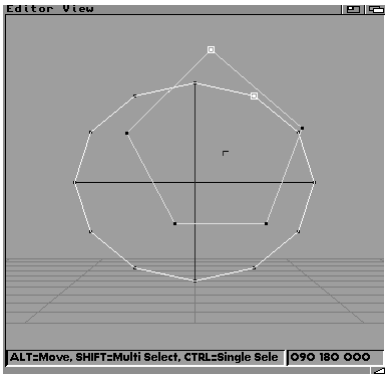
More on Scaling

Scaling of the polygons is very simple to accomplish, even as they also move and rotate, all with the Edit Path requester. Choose Edit Path and look at the gadgets labeled "Scaling." As in rotation, there are specifiers for each of the X, Y, and Z axes. The Entry/Exit gadgets by default set to apply no scaling. If you change Exit to 0.5 for any axes, then during the member's time, the assigned polyspolygons (and any

hierarchy from this point down) will "shrink" to 0.5 - half - of their size. If you change the settings to 5.0/0.1, the assigned polyspolygons will start at 5 times the size at which you've drawn them, and end at one-tenth their size. Picture such an action occurring as an object moves into the frame of view of your animation. The effect would be dramatic. Try these settings. The poly you are scaling is flat in the XZ plane, so it has no Y dimension. Any changes you enter for this dimension will have no effect on this poly. You'd need a poly with a Y dimension to have any effect. The scaling values are multipliers, so polyspolygons must have some initial value. You also use CSplines to control scaling, as with rotation angles, individually for the three axes.

When you set up a complex animation using paths with rotation, scaling, and even instances (but not deforms), it's often a good idea to create a new space and use single polygons to represent the objects you're going to animate. Use a single polygon in the same place in space as the different parts of the objects you are going to move. Set up your paths completely using these single polyspolygons. This allows you to use Preview the motions in real time. When you're completely satisfied with the motion, you can select all the paths, clone them and jump them to the space which contains your full objects, then Assign Path as needed.

Deforms



Now that you can rotate, translate and scale the polygons in any way you choose, what if you want to specify a new position for the polygons that is not just a simple translation? ? This is the purpose of Deforms. Let's do a simple example to see how they work.

Select New and Flat View in the Y axis. Use the Freehand tool to draw a polygon that will represent a path. Use five points similar to those shown in the illustration. Use the Make Arc tool to make a circle (360 degrees) of 12 segments. Select the five-point poly and turn it into a path. Don't specify any movement, rotation, or scaling. Select the circle and assign it to the path. Now for the deform. Select the path and then select the Path menu item Deform / Begin Level. The program has made a copy of the circle and stored it in the path. You cannot see this copy, but it is there. Select the circle and move some of its points. Set the polyspolygons, then select the Path menu item Deform / End Level. You will see the circle (now a starlike shape) return to the shape it had when you began the deform. The path has recorded the first Deform Level and can change the circle accordingly during an animation. Let's see this. Hit <F9>. In the preview you will see the circle change to the star.

Deform Level Editing

Return to the Editor and select the path. Select the Path menu item Deform / Edit Level. The Edit Deform Levels requester opens. Change "Cyclical" to "Periodic," accept and preview. The Deform sequence now occurs from start to finish and back to start. Return to the Editor, and let's set up another level of Deform. Select the path. Select the Path menu item Deform Begin Level. The polygon assigned to the path will jump to the last deform position in the list, in this case, the star shape. Change the star in some way. Move its points, resize it, move it, whatever you want - except for rotation (Deform is straight line based. and will not do rotation properly). Once you're satisfied with the new position and shape of the star, select the Path menu item Deform / End Level. Hit <F9> to preview. You do not see the second deform level because the objects are time lined and you haven't given it any time yet (though you might see it for a single frame if the frame count is just right). Return to the Editor, select the path and select the Path menu item Deform / Edit Deform Levels. The Edit Member box determines which Deform Level you're working on. You can step through the available levels with the forward and reverse (">" and "<") gadgets next to it. Click the gadget to get to Level 2, and click the Even gadget. The time is now evenly divided between the two available levels. Accept and preview. Now that the second level has been assigned some time, you will see it.

You can also use a CSpline to control the time Deform takes to complete. The CSpline is local to the member listed in the Edit Member gadget. The levels react to the path's global variables. This means you can use a CSpline on the global control for the path, and it will control all Deform levels.

Aladdin 4D's Deform feature is similar to what is referred to in some software as a "keyframe" animation system. This type of animation has one major drawback: It's memory hungry. Each deform level maintains a record of every point in an object. When using Deform, keep an eye on available memory. You can set up a beautiful animation that you can't render because of a lack of memory!

Path instructions are similar to procedural textures. They don't require much memory, but they do require some thought to set up. Deforms are similar to bitmaps. They require plentiful memory, but not as much thought.

~~Remember,~~Remember that Deforms can exist simultaneously with the rotations, scaling and translation you have already tried. You may want to use Edit Path to turn on movement, and a few rotations while you are at it, to see all this work at once. It can be astounding.

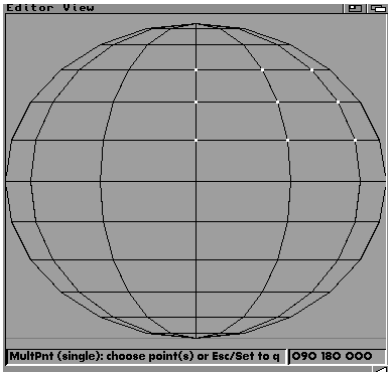
Point Mode

The toolbox's Point Mode gadget puts Aladdin 4D into Point Mode, so you can perform point-wise operations on a number of points at one time. This mode uses "permissions" to choose which points are to be worked on.

In Point Mode, a point cannot move unless it has permission. This applies to all tools that know about Point Mode - scale, rotate, conform, etc. You establish permissions to move in point mode using the Multi-Point_a external tool. You can read about this tool in its About file.

Say you have a sphere on screen and you use the Multi-Point_a tool to give only the top half of the sphere's points permission. Then you enter Point Mode. You will see single pixel points painted in complementary colors on those points that have permission. If you select the sphere and try to move it, Oonly those points that have permission will move. This means, of course, that you can perform actions

on the group of points without affecting the rest of the object.. The points that do not have permission will not be changed.



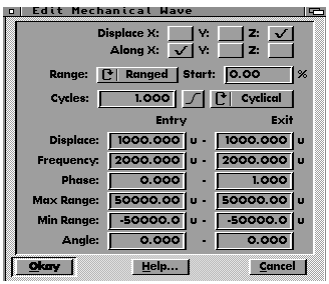
This gives you a great deal of power in choosing those areas that will be affected by your tools, and those areas that will not be changed.

By selecting the Point Mode gadget again, program operation resumes its normal operating mode.

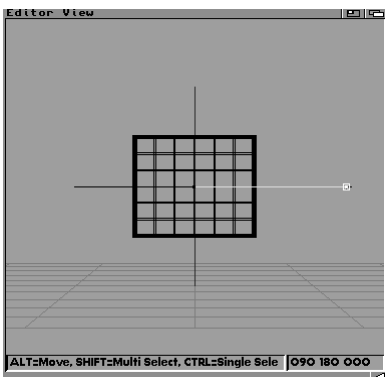
The Point Mode changes the operation of almost all of the tools that change a point's location.. If you wanted to conform only the nose of a face - a Pinnochio remake, perhaps? - you can give permission to only the points making up the nose and then still select the entire face.. The conform will choose its distances based on all selected polyspolygons, but only the nose will move.

CAUTION: This mode causes extremely non-planar (not flat) polyspolygons.. Before rendering you should change the polyspolygons to triangles.. Also, you should be particularly attentive to Shading Groups when you change an object in this mode.. See the tutorial on shading groups.

Mechanical Waves



To see the abilities (and limits) of mechanical waves, select New and flat view in the Y axis.. Use the Make Rectangle tool to make a cube 10000, 10000, 10000 centered to the origin, with the grid option ON and using six segments in each of X, Y, and Z.. Set the polyspolygons.. Put the Attach Point at the origin.. Select the Freehand Poly gadget and draw a straight line out to the right so that it extends beyond the edge of the cube, making it easy to select.



The first point of this line is the point at the origin, which is at the center of the cube.. The first point (point 0), as in rotation, translation, and scaling, is where the waves will originate (which means it will move if translation is turned on, along with the assigned polyspolygons).. This line will be the path that creates the waves, and the cube will be the object that we will "wave".

Select the line and use the Path menu item Make Path.. When the Edit Path requester opens, click the Waves gadget at the bottom.. The Edit Mechanical Wave requester that opens is discussed more thoroughly in the references sections.. At the top are Displace and Along gadgets for each of the axes.. Click Displace Z and Along X.. Leave the other defaults and accept both requesters.. Now select the polygons in the cube.. Note that they are alternately grouped, so make sure you select both groups (use <Shift> to select the second one).. Assign them to the path and preview the animation at 30 frames total, from 1 to 30.

If you inspect the movement closely., you'll see that the polygons of the cube are being moved in a sinusoidal wave.. The displacement is along the Z axis, and the wave crests move along the X axis.. The source of the wave is at the center of the cube, and moves outward in both directions along the X axis from there.. Use <Esc> to get back to the Editor, and Edit Path.. Click Wave again, and select the Along Y gadget (so both X and Y Along gadgets are selected).. Accept both requesters and preview.. The wave

you see is not very apparent.. It looks a mess!! The problem is that there are not enough points in the grid to show the detail of the wave.. We could just replace the cube with one with more segments, but instead lets change the wave.. Get to the Wave requester again and change the frequency to 10000.. Preview the animation again.. The wave is circular (actually cylindrical).

Open the Wave requester again and turn on Along Z also.. Preview this.. All three Alongs are being used to influence the displacement.. The results are nearly identical to using just the XY Alongs in this case.

Try other combinations of Along and Displace.. You can produce linear, cylindrical, or spherical waves by using one, two, or three of the Displace gadgets.. The Along gadgets allow you to choose which axis (axes) measure the distance of the point in the poly to the source of the wave against your range and frequency.

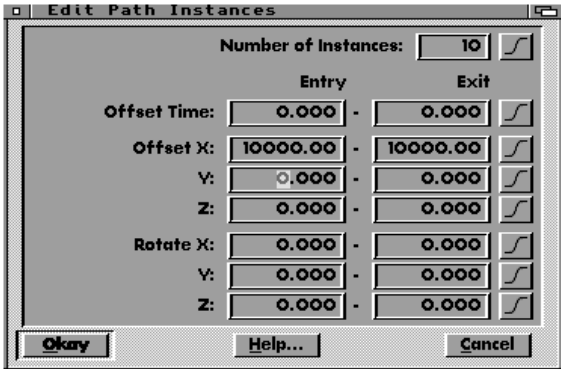
Very Important: In general you should use a grid-like object when you generate linear wave displacement, and a concentric-like object when you generate circular wave displacement.. If not, and the displacement is relatively high, you can experience alternating dark-light problems when animating shaded polygons.

To illustrate a concentric-like object, use the Prim-Q tool to make an elliptic cone with Radius D of 0.0, N1/N2 of 2.000/2.000, and radii of 10000.. Then in Group Level 2, delete the top half of the cone.. Scale the bottom half to 0.000 in the Z axis.. The Wave Source goes at the center and displace the Z along the XY, for a cylindrical wave movement from the center of the concentric rings of polyspolygons.. Other types of concentric-like objects are simple lathe operations.

When using mechanical waves you almost always have to change the objects into all triangles to render, since the waves nearly always make the polyspolygons non-planar if they have more than three points.

Also note that the waves are generated in the directly assigned polyspolygons AND all polyspolygons further down the hierarchy.. This is just as in scaling.

Instancing



Instancing is a very powerful feature that must be used with some knowledge of your computer's limits. To experience the option, we will start very simply, but stick around - the results are spectacular!

Select New, and Flat View in the Y axis. Use the Make Arc tool to make an arc of 360 degrees with three segments (an equilateral triangle, in other words). Set the triangle then select it and scale it to 10 percent of its original size (10.000, 10.000, 10.000) and Set it. Put the Attach Point at the origin (RMB over the Page Move gadget). Draw a freehand poly out to the right and Set it. This will be the path. Select the straight line and choose the Path menu item Make Path. When the Edit Path requester opens, click the Instances gadget. The Edit Path Instances requester that opens is discussed in detail in the references sections. The "number" gadget shows the number of instances of the original to make.

If the number is 0, there are no instances created. Change it to 10. The Offset gadgets define the time and distance over which to distribute the instances. Change Offset X to 10000 for both Entry and Exit. Accept the requesters and assign the triangle to the path. Preview the animation. You will see that there are now eleven triangles, the original plus the ten instances, even though the Editor window still shows only one.

Go back to the Edit Path Instances requester and find the Rotate gadgets. Be sure you're in the Instances Control requester, not the Edit Path requester. Change Rotate Y to negative 360 degrees on Entry and Positive 360 on Exit. Change the number to 60. Preview this. Festive!

Open the Edit Path Instances requester again and make the X and Z Rotate match the Y Rotate gadgets and preview it. Some very interesting effects can be achieved with just one small triangle being instanced many times. The new ~~polys~~polygons are rotated around the first point (point 0) of the path, so the farther they are, the more pronounced the effect of the rotation. The rotation is also divided among the instances, so the original poly is never rotated, and the first only 1/60th (if there are 60 instances) and so on. You may want to change the position of the original polygon relative to the path to see the effect this has.

VERY IMPORTANT:

Instancing creates copies of polygons and so requires memory. When using instances, remember to keep the number of instances small if the number of ~~polys~~polygons to be instanced is large.

If you're instancing only one poly, or a small object like a tetrahedron (four ~~polys~~polygons) you can use relatively large numbers of instances without troubling yourself about memory supply.

Instancing has some useful applications. Let's try a few of these. Set the Attach Point to the top point of the triangle. Now draw a short straight line out from there. Select this new line and choose the Path menu item Make Path. In the Edit Path requester, enter Entry/Exit Y rotations of -30.000, 30.000 and turn the Y rotation type to Periodic. Accept the requester. Select the triangle and assign it to the new path. Then select the new path and assign it to the old path. Finally, select the old path and edit its instances again. Change the number to 10, and change all the Rotates to 0.000 both Entry and Exit. Accept this and preview the animation. You will see that instancing not only instances the polyspolygons assigned, but also all paths assigned and their polyspolygons. This means you can have an involved object, like a bird flapping its wings, flipping its tail, rolling its eyes, etc. and clone it as a whole just by assigning the paths that control its polyspolygons to another path that instances them all.

There is something even more interesting. The instances can have their times promoted or "offset". Open the Edit Path Instances requester for the old path again, and change the offset time to 1.0, both entry and exit. Preview this to see the effect. There are no limits to the time offset. Try 2.0 on both entry and exit. You may want to zoom in and increase the instances to 20 or 30 to see this better. Try -3.0 on both. Try -3.0 on entry and 3.0 on exit!

A useful example of using instancing with its time offsets might be a "train" effect. Delete the small new path. Use Make Arc to make a circle of 24 segments with a radius of 10000 units. Select the circle and choose the Path menu item Make Path. Turn ON Movement Status and Last Poly. urn ON Align, Align Reverse, and Align Last, and set the Align axis to the Y axis. Accept the requester. Now set the Attach Point to the top point of the circular path (point 0). Select the triangle and center it to this point (RMB on Page Center gadget). Assign the triangle to the circular path, and then the circular path to the old path. Edit the old path instances. Use a time offset of 0.5, Entry and Exit, and no offsets or rotates. Preview.

The old path has created 30 (or whatever number you set) circles in the same positions each with their triangles, and offset the times of each circular path. You may want to change the positions of the points in the circle and edit it so the velocity is constant to see a less predictable version. You can also delete the circle, use the Freehand Spline tool to draw a more interesting path, convert the spline to a poly, and convert the resulting poly into a path. Much more interesting! ! Notice, however, that the Edit Path Instances requester offers direct CSpline controls for ALL of its elements.

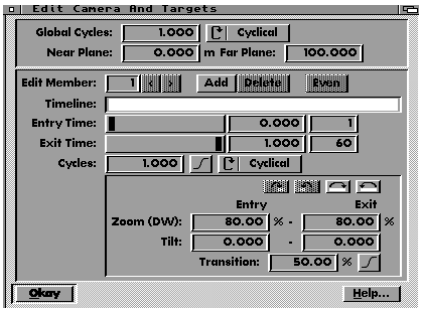
Another useful but not readily apparent use for Instances is to make "rotational clones". Select New and Flat View in the Y axis. Use the Make Rectangle tool to make a simple cube, 10000 by 10000 by 10000. Move the Attach Point to the origin, or center of the cube. Draw a straight line out from there to use as a path. Make the straight line a path and set its instances to 5, rotate Z to 360, both Entry and Exit. Assign the cube to the path. Rotate your view so you are looking down slightly on the top of the cube and preview. You will see what appears to be three cubes rotated around each other's centers. But you asked for five instances (total of six when you add the original). Go back to the Editor and select the cube and move it off to the left. Preview again. Now you see six cubes, rotated around the first point of the path.

Instances always make identical copies of the assigned polyspolygons, including their textures, attributes, shading, etc. For many purposes, it is better to actually clone the objects and position them, especially if you would prefer to have different textures on them.

Also, Instances must be built by the program before previewing or rendering. This can cause a considerable delay if you are using a large number, or the objects being instanced are complex. This delay will occur when they are being built (on entry to preview or render) and when they are being freed (on exit from preview or render).

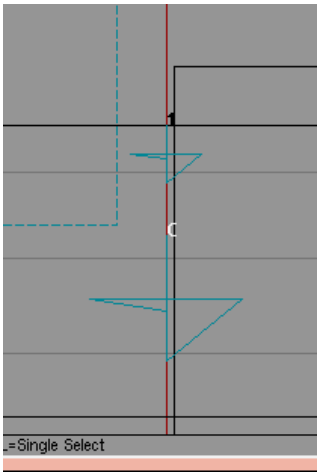
Again, because of memory considerations, remember to keep numbers reasonable.

Camera



The next type of animation to explore is moving the camera and targets in the environment. Let's do this.

Select New and Flat View in the Y axis. Use the Make Rectangle tool to make a cube of 10000, 10000, 10000 centered to the origin. Use the Clone tool with settings of X and Z at 0, Y at -20000. Select and Clone the cube five times.



Set the cubes. Move the Attach Point to the origin. Select the Object menu item Camera. The requester that opens is the Edit Camera and Targets requester. If you already have a camera in the current space, you'll edit the camera. If you select this menu item without having a camera in the current space, Aladdin 4D creates a new one for you, along with its first target (at the current Attach Point). Either way, this requester edits the camera and its targets. For now, just accept the requester's defaults. The camera and its target appears in the Editor window. Flat View in the Z axis and turn ON Isometric View. Zoom out until you can see the back cube at the top of the screen. Select and then immediately Set the camera. This moves the Attach Point to the camera. Select the FreeHand gadget and draw a poly similar to the one in the accompanying illustration. We'll make this poly a path.

Set the new poly. Select it and choose Make Path, turning Movement Status ON and Last Segment ON. Accept the Edit Path requester. Select the camera and assign it to the path. Hit <F9>.

If you adjust the translation of the path, the camera will conform. Any rotation, scaling, deforms, etc. of the path are ignored by the camera. Only translation is referenced in moving the camera. The direction the camera points is controlled by the target. No matter where the camera moves, it will always point to the target. Let's move the target. Select the target and Set it. The Attach Point is now at the target. Use the Freehand tool to draw a path for the target to follow. Select this new poly and turn it into a path with Movement Status and Last Segment ON. Set it, then select the target and assign it to the new path. Hit <F9>. You will see the camera moving along the path and at the same time tracking the target as it moves along its path. Don't forget that you can also assign the cubes to paths, having them bounce, rotate, move, and deform, with the textures on their surfaces changing. You can also use additional targets, and change the zoom level and tilt of the camera (under target control), and on and on. You can use additional targets each altering the camera's view direction at its assigned times, and panning to the next one smoothly or abruptly, all under exact control.

Camera Transitions with Multiple Cameras

Another word or three is in order regarding the use of multiple cameras and camera transitions.

First, notice the time line gadgets in the Edit Camera and Targets requester. Zoom and Tilt are defined at the bottom of the requester, too. The target instructions, zoom, and tilt, are averaged, moving from entry to exit values BEFORE the transition begins. A CSpline gadget offers immense control over the transition time, and another one (Cycles) provides the usual control over the cyclic or periodic time line cycles.

The transition spline controls the rate of change between the exit settings and the entry settings of the next target. If there is only one target, the transition CSpline is ignored. To prevent abrupt jumps in value, the transition CSpline does NOT change from the actual exit values to the next entry values, but instead works from CSpline-ALTERED exit-entry values. The last target transitions to the first.

The transition CSpline also controls the positional change, meaning that camera direction is controlled by this CSpline during the transit period.

The transition ALWAYS begins at the specified position. If, for example, there are two targets with even time, and each has a transition of 0.5, the first target will be active from 0.0 to 0.25, then its transition from 0.25 to 0.5, then the second target from 0.5 to 0.75, then the second transition from 0.75 to 1.0. This is not subject to a CSpline control of its own, so the transition and time bar always hit as specified. The CSplines only change the rate of change in the pre-transition and the transition periods.

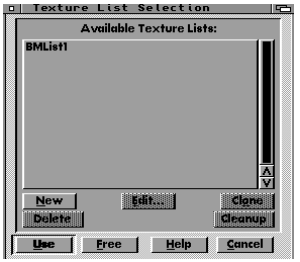
A "jump" will still occur if the transition CSpline does not start at 0.0 and end at 100.0. A "jump" will also still occur in the pretransition time if Cycles is set to a number other than 1 and if Periodic isn't ON.

Also, the cycles and Periodic flags only apply to the pretransition time. The transition only occurs once, regardless of the number of cycles. It can still, of course, be made Periodic if desired, through the use of a CSpline.

Animation tutorial conclusion

That concludes the animation tutorials. There are more tutorials, here, but we have covered most of the basics - and a little bit of the specifics. From here you should be able to do almost anything you want. Do some general playing with the program, to make sure you have the basics down pat, before going on to the rest of the tutorials. Enjoy!

~~Textures~~ Tutorial ~~(Textures)~~



The ability to map textures onto objects is an important facility in any 3D program, and the texture abilities in Aladdin 4D are comprehensive, easy to use, and powerful.

At the heart of the program's texturing capabilities are its Texture Lists~~.~~. In truth, these are not really lists~~.~~. All the other lists you have used (or will use), like the attribute lists, gas lists, wave lists, target lists, deform lists, animation rotation lists, and so on, are true lists~~.~~. They are composed of separate members that are active in turn at specified times~~.~~. They are lists because the time period of one member must not overlap any others~~.~~. Texture Lists are fundamentally different in that the members can overlap in time~~.~~. This is what is sometimes referred to as a random access stack, but for simplicity we have called them "Texture Lists."

In Aladdin 4D you can apply single texture, just as you would in any other 3D program~~.~~. However, you can also combine any number of textures on the polygons for more complex, realistic (or unrealistic if you wish) effects~~.~~. These textures may exist all at the same time, or they can suddenly appear - bang~~!~~! Or they can fade in slowly~~.~~. And back out~~.~~. They can change from full color to grey or anywhere in between~~.~~. They may even fade from color to bumpmap to illumination to specular maps, and on and on.

Think of the Aladdin 4D's Texture List as a series of LAYERS of textures~~.~~. The first layer is the base color of the poly, as established in the Attribute List (which of course, can also change in almost unlimited ways)~~.~~. After that, the second texture layer is the first one in the Texture List~~.~~. The third layer is the second texture in the list, and so on.

This brings up some questions: Won't the second texture just cover up the first one~~?~~? The answer is, only if you want it to~~!~~! There are several methods of allowing previous layers to show through~~.~~. First, if the texture is a "bump" texture (and has no color contribution of its own), it never covers, but only modifies~~.~~. Indeed, two or more bump textures can be combined and influence each other, though some limits exist on bump maps for transparent ~~polys~~polygons~~.~~. Specular (hardness) mapped textures and Illumination maps behave in a similar manner.

Second, you can use a Strength setting of less than 1.0, to make textures work over each other~~.~~. If the Strength value of the texture is 0.5, then when it is applied the final color will be half of the color in the texture and half of the underlying color~~.~~. Third, you can specify a special "Decal" texture type~~.~~. This texture contributes no color where the color of the applied bitmap is Color 0 in the picture's palette. (For 24-bit bitmaps, color 0 is defined as the color of top left pixel in the bitmap.) This special Decal mode is the normal mode for Procedural textures~~.~~. When you set up a Procedural you can choose how the colors are arranged, including open spaces where previous textures will show through~~.~~. Of course, you can even use a Decal texture at less than full strength, and there will be a transparent glazing of texture in the areas that do not have color 0.

Aladdin supports several other types of texture mapping, including "Genlock." We'll come to some of these in this tutorial, but they're all detailed in the reference sections.

All Aladdin 4D textures can animate~~.~~. Bitmaps can be single-frame sequences~~.~~. You can also position the bitmaps, allowing you to pan, invert and zoom them during an animation, even if they are only a single picture~~.~~. Procedurals allow you to change not only their color, but the roll amount if they use noise, and all aspects of their spacing and arrangement.

There is even a convenient "negative" switch in the Texture List, so if you want the texture to be applied as its negative image, just turn ON this switch, which inverts normal, bump, illumination, opacity, etc.

Are you excited yet? Let's do a few introductory examples first, then you can let your wildest ideas fly.

First we need something to see textures on, so let's start with a simple flat rectangle. Flat View in the Y axis. Select the Edit menu item Make Rectangle with X of 20000 and Z of 20000. Accept the requester. Next we need an Attribute List for the rectangle. Select the rectangle and select the Object menu item Attributes. This will open the Attribute List requester. Load the list called "gold" and accept the requesters. The rectangle will use this list of attributes during render. The rectangle is still selected, so select the Object menu item Shading. Turn ON Phong and accept (It doesn't matter whether Smooth Adjacent is on, since there are no adjacent polyspolygons). Set the polygon and render, if you like, to see what's there so far.

So far w, we have a simple, gold-colored, Phong shaded rectangle. No big deal, b. But useful in demonstrating textures.

Procedural Textures

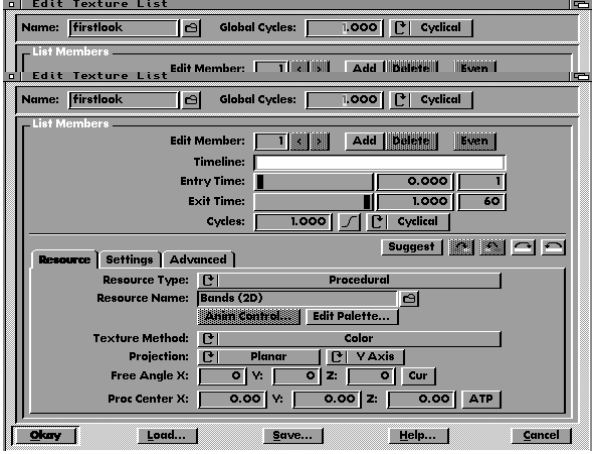
Screenshot: Texture List requester /j

Select the rectangle and select the Object menu item Textures. The Texture List Selection requester opens. Click the New gadget, and the Edit Texture List requester opens.

Rename the Texture List "first look". The List Members area of the requester is where you really control the object's appearance over time. To work on a particular texture in the named list, just get its number in the Edit Member box. You can step through the available members of the texture list, adjusting each one as you choose. To change which member of the list you're working on, you can either use the forward/reverse clickable gadgets or simply type a number into the box.

The default texture is a procedural texture called "bands(2D)". This is the default texture of all brand new Texture Lists. Just accept the requesters (all three) and you're back to the Editor window. Let's take a look at the textured polygon. Select the Render menu item Render Settings. Turn ON Light, Fill, and Phong, leaving Texture OFF for now.

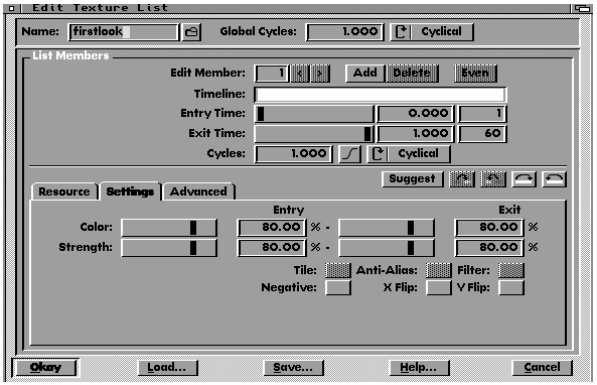
Select the Render menu item Render. As the screen paints you see the gold colored rectangle. This is the "vanilla" version of the rectangle. No texture is visible, because the global Texture permission flag is OFF. Select the Render menu item Render Options, and turn Textures ON. Close the requester and redraw the image. The default texture becomes visible. The Bands procedural texture is just that, a series of bands. It is the simplest of the textures. To demonstrate the basics of procedural textures, we'll alter this texture in several ways



Return to the Editor. Select the rectangle, open the Texture requester and get to the Edit Texture List requester. You didn't have to select the Texture List when you open the first requester because the rectangle is selected, and Aladdin 4D remembers what list is assigned.

Click the Settings tab in the bottom panel and look at the Color and Strength gadgets. They have a range of 0.0 to 1.0, but are currently set to 0.8. This means that the texture's color is set to 80 percent and its strength is at 80 percent. Let's change the color first. Change it to 0.1 (10 percent).

Accept all requesters and render. The image shows that the texture is still there, at 80 percent strength, but its colors are much softer. If you set the Color gadget to 0.0, the texture will be just as strong, but the color is completely removed, using just the Luma in the texture. In other words, it turns the texture into a grayscale. Why would you not want full color in a texture? Well, how about turning a grayscale image into a full color image during an animation? Or how about taming a "hot" color scheme for NTSC display?



Return to the Editor and the Edit Texture List requester (Settings tab) again. Turn the color up, all the way to 1.0. Move Strength down to about 0.2 (20 percent). Close the requesters and render this. As you probably expected the texture is there in full color, but with only 20 percent strength. The base color of the poly, gold, is being used for the other 80 percent and is quite dominant.

TIP: During renders, you don't have to wait for the entire image to complete. You can press <Esc> any time after the image begins to appear and you've seen what you wanted.

Return to the Editor and the Edit Texture List requester again. Turn the strength of the texture all the way up to 1.0. Let's look at where the color comes from.

Select the gadget labeled Proc Colors. This is where the colors for the procedural textures are chosen.

In the upper left corner of the requester is a gadget labeled Colors currently containing "8". This is the maximum number of colors permitted for a procedural. Change this to 3, accept all requesters and render. As you can see, only three colors are now used to build the texture. The rest of the polygon is unaffected by the texture, and is the specified gold color. Return to the Editor and get to the Edit Palette (for Procedural colors) requester again. Look at the color swatches. There are two sets, one for each of Entry and Exit. The available three colors are grouped to the left of the color bars. Near the bottom of the requester is a gadget labeled "Even." Click it, and the Entry colors spread out in the spacing area. Click Exit and then click Even again. Accept all requesters and render. The colors completely cover the polygon. Return to the Editor and get to the Edit Texture List requester (not the Edit Procedural Colors requester).

~~/// FIX 0415 0504 /j~~

~~/// There is no gadget named Width as described 0415 0504 /j~~
~~@In the left~~Under the Advanced tab, in the right columns of gadgets, the top one is labeled Width~~.~~ It reads 10000.
~~@~~If you remember, this polygon is 20000 units wide, and 10000 is the width~~of~~
~~@the~~of the color spacing you just made even~~.~~ When you rendered,~~the~~
~~@colors~~the colors covered the polygon from the center to the edge, a distance of~~10000.~~
~~@Change~~10000. Change the width to 5000~~.~~ This means the colors appear twice~~between~~
~~@the~~between the center of the poly and the edge~~.~~ Accept the requesters and render.

The texture is mirrored at the center of the polygon, the center of the procedural texture~~.~~ It can be changed~~.~~ Return to the Editor and get to the Edit Texture List requester.

In the bottom panel, when the Resource tab is active, there are gadgets labeled Proc Center for each of X, Y, and Z axes~~.~~ There is also a gadget called ATP.

If you click the ATP gadget, the coordinates of the current Attach Point are entered for the center of the texture~~.~~ Try this~~.~~ The display will change depending on which corner of the rectangle is the Attach Point~~.~~ Accept the requesters and render~~.~~ The mirroring is now eliminated~~.~~ Really, this mirroring still occurs, but it's no longer on the polygon~~.~~ The center positioning for a procedural texture can have an important impact on how the texture appears.

Return to the Editor and get all the way to the Edit Procedural Colors requester~~.~~ Two gadgets near the top of the requester are named L.Blend and R.Blend, with spaces to enter percentages (default: 20.00)~~.~~ Change these to 100.00~~.~~ This is the amount of blend between colors that the texture performs~~.~~ Accept and render the image - the colors blend this time~~.~~ If you're using a HAM screen, you may have to do some palette matching, as you did with the spheres in the shading tutorial, to see this one clearly~~.~~ Notice that Blend creates many more than the three colors we've specified~~.~~ This blending also helps eliminate jaggies, although this image doesn't really show it off~~.~~ You can establish separate Left and Right blends choose for each color that you're using, if desired~~.~~ When you select the Blend gadget and enter a number in the requester, it changes ALL blend values for all the colors in the currently selected area, either entry or exit.

Spacing

Let's try one more thing here: Spacing~~.~~ Select the second color (white) in the entry area~~.~~ Look at the slider gadgets labeled Start % and End %~~.~~ Move the one marked End to the left~~.~~ The spacing area shows the change~~.~~ You have opened up a space between the red and white colors and made the white color smaller~~.~~ Can you guess what this will look like when rendered~~?~~ ? Try it and see~~.~~ As you probably guessed, the gold color of the polygon was revealed in the open area between the white and red~~.~~ Think about what would happen during an animation if the color spacing on the entry is different than the color spacing on the exit~~.~~ The animation will show the color spacing changing smoothly between the two spacings you have chosen.

~~/// ??? There is no Proc. Orientation gadget. What takes its place ??? /j~~
~~/// This appears to be "Free Angle" projection type and Free Angle XYZ~~
~~/// now. Is this correct ??? If so, this will take some explaining 0504 ??? /j~~
~~@~~Return to the Editor and open the Edit Texture List requester. ~~On the left side~~
~~@there is a gadget labeled~~Change the Proc~~.~~ Orientation setting~~. Change this~~ to 90.0~~.~~ and render
~~@~~the image~~.~~ As you can see, this is a modifier to the angle that the

@procedure is applied.

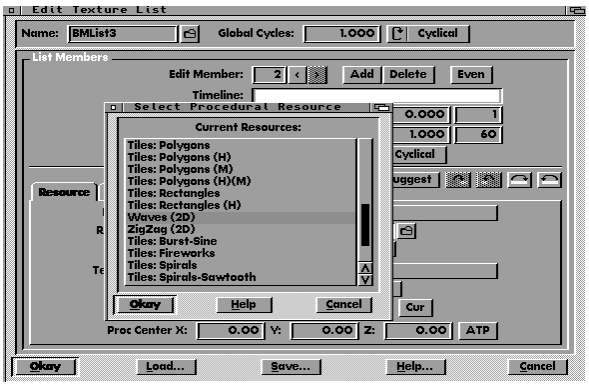
Return to the Editor window. This time we'll add another member to the Texture List. When you add or delete a member from a Texture List, all polyspolygons using that list must be selected when you make the change. Select the rectangle and get the Edit Texture List requester again. The list announces at the bottom of the requester that it is enabled. If you didn't selected all polyspolygons using the list, the requester reports "disabled"
and only allows you to change non-critical values such as Strength and Color. Click the Add gadget at the top right of the requester. The Member gadget now reads "2". You have created a second texture in this list, identical to the first.

/// FIX: What is Proc Orientation renamed? 0415 /j
@Change "Proc Orientation" for this Texture Member to 0.0 and render
@the image. As you see, the first texture is on the polygon with the gold
@showing through, and the second texture overlays these, with the first
@texture and the gold poly showing through its open spaces.

Return to the Editor and get the Edit Texture List requester again. Move to the second texture and change its Strength to 0.5 or thereabouts. Change Proc Orientation to 30.0. Open Procedural Colors, and click the Even gadget. Close all requesters and render the view again. Is this what you expected? ? The mirrored area may be visible again for the second texture, and the member's center would have to be moved to change this.

Return to the Editor and open the Edit Texture List requester again. In the middle of the requester is a "Time line" gadget. It's filled in now, showing that the currently selected Texture List member is active throughout the animation. Use the slider gadget marked "Entry Time" to move the starting time for the member to about 0.2 (20 percent). Close all the requesters and render the image. You see that the image, which is the first frame of the animation, has only the second member on it, since the first member isn't active yet. You can "flash" textures in or out using this technique. You can tell the program that a texture doesn't become active until a certain frame is reached, and at that time has a strength of 0.0, then at the end of its time has a strength of 1.0 - so the texture will wait till its time, and then gradually fade in. Fade-outs are accomplished similarly. Of course, as an alternative, you can have a second texture change strength from 0.0 to 1.0 and the base texture not change at all, which will appear to fade the second one in and the first one out, even though the first one doesn't change. Also, each member can be placed under CSpline control for complete control over the rate of change and delays.

So far, you've experienced many aspects of procedural textures. There is much more that you can do, but there are also other textures. Let's take a look at a few of them.



Before continuing, however, understand why the rectangle had to be selected for the last change. There are certain changes you can make to a texture list without selecting the polygons, and there are certain changes for which you must have polygons selected. The reason for this is that when the polygons are selected, the Texture List is assigned to them when you close the Texture List requester. The polygons need to have a little memory allocated and some information recorded in it for each member in the Texture List. If you add a member and don't assign it, the polyspolygons don't have this memory for this member, so it doesn't work. Some of the changes you can make will also change the information

recorded for the ~~polys~~polygons, and if you change this, you need to have the ~~polys~~polygons selected (also called reapplying the texture)~~.~~. Most of the features in the Texture List can be changed without reapplying the texture~~.~~. The exceptions are:

- Adding or deleting a member;
- Changing the map type (i.e~~.~~. projection, wrapping, etc.); and
- Changing from one texture to another.

The Edit Texture List requester will tell you If you open it with only part of the assigned ~~polys~~polygons selected~~.~~. Aladdin 4D will also inform you if you choose a list with unselected ~~polys~~polygons.

Most other features of Texture Lists can be changed without reapplying the Texture List, so you can still open the Edit Texture List requester and make these changes, even if you have not selected all ~~polys~~polygons that use the list.

~~#### screenshot: Texture List "wave" /j~~

Let's take a look at a couple of procedural textures that are modifications of the "bands" texture~~.~~. Return to the Editor - we're going to change to a new Texture List~~.~~. Select the rectangle~~.~~. Open the Texture List requester and select New to add another list~~.~~. Name it "second try"~~.~~. In the Edit Texture List requester's Settings tab,, set Strength and Color to 100.0 percent for both Entry and Exit.

Click the rotary Resource Type gadget till it reads "Procedural"~~.~~. The gadget underneath it specifies the actual texture file to use~~.~~. If there's nothing there, or when you click the Load/Save gadget next to it, the Select Resource requester opens~~.~~. This is where you can change the texture in use~~.~~. The names of the procedurals you can choose appear in its upper window~~.~~. Currently, the "bands" texture is selected~~.~~. Scroll to the bottom of the list~~.~~. Select "waves" and accept~~.~~. The texture listed has changed to "Waves"~~.~~. The gadgets under the Advanced tab in the Edit Texture List requester ~~adjusts itself~~adjust themselves as necessary to support elements that apply to the texture that's being edited~~.~~. The "Bands" texture only needs a width setting, But this~~Waves~~

~~/// There is no gadget named Width as described. 0415 /j~~

~~@texture~~Waves texture uses Width, Height, and Amplitude~~.~~. You must enter values for these under the Advanced tab.
~~/// ??? 0415 0504 /j~~

Aladdin 4D maintains a lists of defaults for the parameters in its requesters, and when you select something, it puts these in place~~.~~. They will remain until you change the Settings and save them for any requester~~.~~. You can use the requester with non-default settings, and have its defaults come back next time it's invoked - click the "Use" gadget, if there is one~~.~~. Accepting the requester stores your new settings as the new defaults.

In this case just use the selected texture's defaults~~.~~. Accept the requesters and render the image~~.~~. As you can see, Waves is derived from Bands~~.~~. Everything you learned about Bands' color spacing, blending, orientation, etc~~.~~. will also be useful for this texture~~.~~. You may want to try a few alterations before going on.

Return to the Editor~~.~~. Get to the Edit Texture List requester~~.~~. We're going to change from one procedural texture to another, so select the rectangle~~.~~. Open the Select Resource requester~~.~~. The Waves texture is still selected~~.~~. Select the Scallops texture and OK the requester~~.~~. The defaults appear automatically~~.~~. Close all the requesters and render~~.~~. As you can see, the Scallops texture is like the Waves texture, but only positive parts of the wave are used~~.~~. Return to the Editor and change the texture to the ZigZag

texture. Close the requesters and render the image. As you can see the ZigZag texture is also related to Bands, but implements a sawtooth pattern.

Now let's take a look at Noise based textures. Generally, we think of noise as being unwanted, but in rendering many surfaces in a "natural" way, we can come much closer to the realism we seek by introducing controlled noise algorithms in texturing. Noise, in this case, isn't unwanted trash, but carefully sculpted variations in surface regularity that can simulate "real" textures.

Return to the Editor and get to the Edit Texture List requester again. Click the rotary Resource Type gadget till it reads "Procedural". When you click the little square gadget next to it, the Select Resource requester will open. This is where you can change the texture in use. Select the Noise Open texture and accept the requester. Click the Cur (Current Attach Point) gadget. Close the requesters and render the image. As you can see, the texture is being applied according to a fractal type of noise. This noise pattern is quite large - let's make it smaller. Return to the Editor and get to the Edit Texture Listrequester.

@requester. Click the Advanced tab. Width, Height, and Depth??? 0504 /j
now read 10000.0. Change them to 1000.0, close the requester and render the image. What a mess! ! You may think this is not useful, but it is actually one of the most useful textures. The problem is in the color selection. Return to the Editor and get to the Edit Procedural Colors requester. Eight colors are in use, and they are not related. The texture we have defined has lots of color changes on an almost pixel by pixel basis. We can't resolve this on the screen, but perhaps we don't need to. We do need the colors to relate to each other. Change Color 1 to a yellow of maximum brightness (255, 255, 0). Click the Spread gadget. The mouse pointer changes to the "TO" image. Click the last color in the entry area (black). The entry colors change to reflect the spread, and are now closely related. Close all requesters and render the image. As you can see, the smaller noise texture is quite useful when the colors relate well to each other. This texture can be used to imitate grass, wood, and granite by changing the size of the noise, the colors, the color spacing, and the projection angle. Go ahead and try the Noise Closed and Noise Bounded textures to see how they differ.

Other interesting procedural textures are the Tiles types. Return to the Editor and open the Edit Texture List requester. Change the resource to the Tiles Burst texture. As you can see, there are quite a few aspects of Tiles textures that you can edit. Open the Edit Procedural Colors requester and change the number of colors to 8. Change the colors so that they are NOT related, this time. Click the Even gadget. Accept all requesters and render. This texture is a rectangular grid, with the available colors being used in a radial spray from the center of each tile. Return to the Editor and open the Edit Texture List requester. Looking at the columns, Width and Height control the size of the rectangles in the grid. Burst is the number of times the color spacing will be cycled in the radial spray. You may want to change this to 3 and render to observe its effect. The next gadget is Frame. This allows you to specify an area around each tile as a percentage of the width setting.

Change this to 0.05 (5 percent). Next is Color. Color is the number of the color that will be used for the Frame. It can be 0 to 8. If it's 0, it specifies a clear area. If 1 to 8, it specifies the color position in Edit Procedural Colors. You probably have black in Color 8, so change it to 8. Blend Distance also modifies Frame, and is the amount of Blend the Frame will use (also based on the width). Change this to 0.07. Finally, Even and Odd Offsets allow you to "slide" the rows of tiles. Change the Even Offset to 0.5 and leave Odd at 0. Close the requesters and render. Try several combinations for these variables. so you'll be familiar with their effects. Try out the other Tiles textures, too.

See the reference section on textures for a complete description of each procedural texture and a description of the options for each.

Did you notice the negative switch for the Texture List members?? If you want a texture to be applied as its negative image, just turn ON this switch.. This inverts normal, bump, illumination, opacity, etc.

So far we haven't tried Cubes, Blocks, or Helix.. These textures operate in three dimensions.. When you try these textures, remember that they are 3D, not 2D and should be used on a full 3D object, not the flat plane we've been using.. The controls are similar to the 2D ones.. Try them at your leisure.

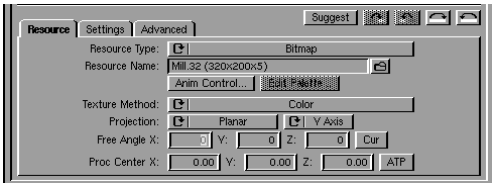
Bitmap Textures

Using bitmaps for texturing is easier than using procedurals.. Part of the reason is that even though you can do as much with bitmaps as you can with procedurals, the approach is less mathematical.. The bitmap you use is image an itself, either drawn in a paint program, or digitized from a photo, video or real life.. So most of the work of specifying shapes, sizes and colors is done when the bitmap is made.. We only have to specify how we want it applied, what effect the lighting will have on it, and how it will change over time.

It is assumed that you have worked through the procedural section and know how to set up your display mode, load an Attribute List, apply shading, and use the Make Rectangle tool.

Let's start with the a rectangle, as in the procedural tutorial.. Flat view in the Y axis and make a rectangle 20000 by 20000 in the XZ plane.. Load the Gold Attribute List and apply it to the rectangle.. Phong shade the rectangle.. Now we're ready for some texture.

Select the rectangle, then select the Object menu item Textures.. Make a new Texture List by using the New gadget.. In the Edit Texture List requester, name the texture "first bitmap".. Click the Resource Type gadget till it reads "Bitmap." Click the load resource gadget to display the available bitmaps.



Aladdin 4D's file requester shows some special "devices" to load from and to save to.. These are for your convenience in keeping the various types of files logically ordered on your system.. The actual location of these "logical" items - like "Aladdin 4D:drawings/" - is controllable in the icon Tooltypes.. Simply edit these to suit the way your system is organized.. If you're loading and saving often, it'll be quite convenient to have these shortcuts to take you to your "stashes" of bitmaps, fonts, etc.

Go to the textures drawer of the examples disk and load the Mill.32 texture.. Select the texture and accept the requester.. The bitmap Resource Name area now tells you some information about this texture.. This is only a 320x200 image in 32 colors, so don't expect photographic results!. Normally you'll use better resolution and probably 24-bit (or JPEG), 256-color, or Ham-8 textures, but for this tutorial this small one will do nicely.

The Advanced Tab contains settingsspecifiers for "First and Last" indices.. First X Index is the percentage horizontally that will be used for the left position of the bitmap.. If you choose 0.0, the left pixel used will be an X of 0.. If you use 0.5, the left pixel used will be an X of half of the bitmap's width, for example, 320 for a 640-pixel-wide bitmap.

Last X Index is the corresponding percentage of the bitmap that will be used for the right side. There are corresponding percentages for the top and bottom pixels (the Y axis in the bitmap) in First Y Index and Last Y Index. The default settings allow the entire bitmap to be shown. Accept the requesters and the texture is applied to the rectangle. Render the image. You will see an image of an old mill on the polygon. Return to the Editor and Edit Texture List. Click the Advanced tab. Change First X Index to 0.5 and First Y Index to 0.5. Close the requesters and render the image. Do you see that the starting X pixel is halfway through the bitmap's width, and the starting Y pixel is halfway through the bitmap's height? You can, of course specify an entry and exit value for these percentages which will allow you to pan, scroll, and even flip and zoom a bitmap during animation. You may want to try making a small animation doing just this.

These indices have no limit. You can specify numbers both greater than 1.0 and less than 0.0! For instance, if you want to see the old mill centered in "bare polygon", use First values of -1.0 and Last values of 2.0 for both X and Y. A smaller border would result with First values of -0.5 and Last values of 1.5, or even First values of -0.1 and Last values of 1.1. See the relationship? The area outside the bitmap is treated as color 0, with all its properties for the different texture types.

The bitmap texture is, of course, composed of pixels. If you're far enough from the polygon, these pixels are not individually visible. But when you use a bitmap texture and have the camera moving, you might get extremely close to a bitmap textured polygon and be able to see the individual pixels in the polygon. This does not occur with procedurals, as they are not composed of pixels, but of programming instructions to the renderer.

Return to the Editor and click the Zoom In gadget four times. Render the image. As you can see, the pixels become quite prominent. Aladdin 4D can compensate, however. Return to the Edit Texture List requester.

In the Settings tab, turn the gadget labeled "Anti-Alias" ON. Close the requesters and render the image. As you can see, the pixels are being averaged, resulting in a "no pixel" look from a decidedly pixelized original texture. Even in this extreme example, the results are quite pleasing. For most of your work you won't encounter such an extreme case, but when you need it, turn Anti-Alias ON to help out. See also super-sampling.

There are two situations in which you'd always use the Anti-Alias option: when you are going to zoom in close to a bitmap, and when you are using a bitmap as a bump map. "Too close" means any time the texture itself would be displayed larger than its original size. This includes using a bitmap as a background when the bitmap is smaller than the background will be. Another use for the Anti-Alias setting is when using bitmaps of limited color depth, like this ~~one which~~one that has only 32 colors. The averaging between the pixels simulates more colors than are present in the bitmap, smoothing it.

(Anti-aliasing OFF) (Anti-aliasing ON)

There is another option in the Settings panel, right next to the Anti-Alias gadget. It's called Filter. Try turning Anti-Alias OFF and Filter ON. Click the Zoom Out gadget four times to get back to the original view and render the image. The image is slightly blurred. The Filter is a standard 3x3 blurring convolution filter. It finds the pixel in the texture that is to be used for the pixel in the render, and then averages it with the 8 surrounding pixels in the texture's bitmap. It is useful when you get far away from a textured poly. Again, this option is only for bitmap textures since there is no need for it in procedurals. Its use is not very common, so it's normally off.

Let's see this one as a bump map. Return to the Editor and click the gadget marked Texture Method. It reads "color", the default. Each time you click this rotary gadgetcycle gadget, a different method of texturing is displayed. Continue clicking until you see "Bump Map." The default settings use the whole bitmap. In the Settings tab, move the Color slider all the way down, to 0.0. Close the requester and render the image. See how the surface of the polygon seems to be embossed with the image in the bitmap? The luma (the largest value of the red, green and blue components) of the pixels in the bitmap is being compared with the surrounding pixels, and the normal of the polygon is being deflected as a function of these differences. The result is that the surface of the polygon seems to be dented, or bumpy. These deflections in the normal cause the polygon's "dents" to actually behave with a passing light source in a very natural manner. The size of the bumps is determined by the strength of the texture and the difference in luma (in other words the contrast level) of the bitmap.

The reason for turning the color all the way down is that the Bump Map texture method uses the color slider in a different way than the Color method. Return to the Editor and open the requester and set the color slider to 50 percent. Close the requesters and render the image. The color in the Color texturing method removes color saturation from the texture. In the Bump Map method, it allows you to add some color from the bitmap used to create the Bump Map. Of course, you could have just added another member using the same bitmap but with Color selected as the Texture Method. However, this double use of the bitmap is far more efficient. This specification for the color is present in Illumination and Hardness (specular) Texture Methods as well.

Bump Maps are not limited to bitmap textures. You can also use a procedural as a Bump Map. Remember when using procedurals as bump maps that higher levels of blend result in smoother bumps. Also keep in mind that the blend type Bell will round the bumps, while the default gives angular bumps.

And finally, the size of the bump is controlled by the strength of the texture and the luma difference between the adjacent colors. Black to white give the most contrast and the largest bumps. Red (200, 0, 0), green (0, 200, 0) and blue (0, 0, 200) all have the same luma and will not reveal a bump.

Bitmap Tiles

Open the Edit Texture List requester again. Change Texture Method back to Color and use full color and strength. Click the Tiles gadget to turn it ON. This is different from the Shingles type or the Procedural tiles types. It is like the procedural tiles, except it allows you to use a bitmap and have multiple images occur. Leave the defaults, accept the requester, and render for a quick look. There's more about this type and its possibilities in the reference section.

Next let's take a look at Reflection Maps. For this we need more than just a rectangle. Let's make a 12 x 12 sphere. Return to the Editor and select New. You can either use the ~~Prim~~ qPrimitive-Quad tool to make a sphere (ellipsoid with N1/N2 of 2.0/2.0) or if you prefer, you can make your own like this: Use the Edit menu item Make Arc to make an arc of 180 degrees. Set the Attach Point to the top of the arc. Select Z Active Axis (so the Lathe will go around the Z). Get the Lathe defaults (RMB on the Lathe gadget). Use 360, Last Segment OFF and Solid ON Accept the requester, select the arc, and Lathe it (left mouse button on the Lathe gadget). Delete the template poly.

Either way you now have a sphere. Select and Phong shade it. Reflection Maps require Phong shading. While the sphere is selected, open the Attribute List requester. Make a new list with the Add gadget. Control this list and name it "Chrome." Get the Edit Texture List requester.

Think about chrome - the metal, not the list. What color is it? ? Chrome surfaces actually have the colors of everything around them. If you look into the polished bumper of a car, (they still make chrome bumpers, don't they?) you see, not the bumper, but the reflection of the driveway, the grass, the trees, the sky and clouds, etc. Chrome itself has no color. The idea is that the bumper reflects nearly all of the light that hits it. But what about a red chrome ball? ? The red chrome ball, on the other hand reflects the red component of the light better than the rest. Both are nearly fully reflective, but their specular reflection is a little different. We can imitate both of these situations quite well. In the Attribute List, there is a gadget called Reflection. This is the amount of reflected of light. It has been separated from the Map Reflection to give you considerable control. The Hardness setting and Highlight Size are also independent of Map Reflection for the same reason.

Set Reflection to 255, Hardness to 255, and Highlight Size to 200. Set Map Reflectivity to 200. This causes the ball to have a hard appearance, and use about 80 percent of the Reflection Map. Use the color sliders to make the red, green, and blue values 0. This means the ball is black, or has no color and no luma. Close the requesters, and the ball will be assigned the Attribute List. Now the texture.. With the sphere selected, open the Texture List requester and make a new one with the Add gadget. Get the Control requester and name it "reflect one". Get the Edit Texture List requester. Click the Select gadget and Show Bitmaps. Load the texture called "Mill.32", a low-res 32-color picture. If you have some beautiful 24-bit images you'd rather use, do so, but make sure there is a recognizable figure in it, not just a repeating or random pattern. Select the texture and close the requester. You will see the texture's name and information in the bottom left of the requester. The Texture Method gadget has "Color" printed in it. Click it until it reads "Reflection." Turn Anti-Alias ON.

The requester displays the default values for Reflection texturing. Accept the requesters and render the image. You should see a nearly perfectly reflective sphere, reflecting the image of the old mill. Return to the Editor. Open the Clone defaults and set the X offset to 22000. Close the requester. Select the sphere and clone it. Set the new sphere. Set the Attach Point to the origin (RMB on the Page Move

gadget).-. Select all the ~~polys~~polygons (RMB on the Multiple Select gadget).-. Center them to the Attach Point (RMB on the Page Center gadget).-. Now select the new sphere, on the right.-. Open its Attribute List (which was also cloned).-. Rename it "red reflective".-. Get its Edit Texture List requester and change the colors to RGB values of 200, 40, 40 (a nice red) and set its map reflectivity to 150.-. Accept the requesters and render the image.-. This is a good simulation of a reflective red glass or plastic.

Now let's try using a Wave Source to achieve the same thing.-. Return to the Editor.-. When you add a Wave Source (or light, target, etc.) it is always created at the current Attach Point.-. Set the spheres if they are selected, and set the Attach Point to the origin (RMB on Page Move gadget).-. Select the Object menu item Wave / New.-. The Wave Control requester opens.-. Just accept the requester at its defaults.-. The wave is a small triangle at the origin.-. Select the Wave Source, and using the left side of the screen as a straight edge, and holding the <Alt> key, move it up near the top of the screen.-. Set it there.-. Render the image.-. You can see that the reflection is being altered by the Wave Source.-. You may want to move the Wave Source to different positions and try it, but first let's try a couple of changes.

Return to the Editor.-. Select the Wave Source and select the Object menu item Wave Edit.-. The Wave Control requester opens.-. The Wave Source can have separate members on a time line.-. The specifics of these is discussed in the reference section.-. For right now, find the Spherical Wave Type.-. Click this until you see Multi.-. The locked gadgets become free.-. Accept the requester and render the image.-. Try some different combinations - refer to the section on Multi in the reference sections if you need to.

The Multi option is usually used on flat planes instead of spheres or other three-dimensional objects, but it can be quite impressive.-. Now edit the wave and click the Wave's Type gadget until you see Bumps.-. Render this to see an easy and most useful Wave Type.-. The amount of Reflection Map is actually flattening out the spheres' appearance.-. The dark side of the sphere is not dark, but full of the Reflection Map.-. You can get a more satisfactory image by lowering the map reflection in the Attribute List, and giving the chrome sphere a medium grey color.-. After all, a truly perfect reflector doesn't exist in nature, does it?

Return to the Editor.-. Get the Edit Texture List for the red sphere.-. Look at the column of gadgets.-. Change the Scaler to 4.-. Close the requesters and render the view.-. Observe that the Reflection Map has been used in a multiple fashion.-. The Scaler can be used in this manner for special effects.-. Its usual purpose, however, is to get only the amount of the image you want onto the object, without showing the edges of the bitmap.-. Keep in mind that the screen influence is used to change the index when the object moves on the screen, so you want to make sure your Scaler is not going to allow the edges to be seen even when your objects are nearly off the screen.-. The default value for the Scaler is very good for general use.-. If you look at the member requester again you will notice that you can also change the X and y percentages as you did in the earlier example.-. The Scaler is still enforced and the multiple image will be there, but with a zoom, pan, etc during the animation.-. This can really give you some great effects, like a reflective multi faceted eye on a bug.-. And of course the XY indices still can be used for some really dazzling effects.

One of the most surprising uses for this type of reflectance map is the quick simulation of glass.-. This is so startling, you won't believe it isn't raytraced.-. It is only an illusion, however, and only works with backgrounds and Reflection Maps using the same texture.-. Here's how to do if.-. Load your favorite digitized background image.-. Now apply it to a sphere, or other 3D object as a reflectance map.-. Make a new Texture List, using the image as a normal texture, and use it for the background.-. During the animation, move the object up and down, or left and right on the screen.-. The effect is enhanced if you add a bump Wave Source and move it along with the object.-. Adding some transparency in the object, so the background peeks through it, can also be interesting - the back of the sphere then shows the inverse image, like a lens.-. Another technique is to set the background texture's X percentage so that it has a First X index of 0.0, and a Last X index of 0.5, and the Reflection Map textures X percentage so that it has a First X index of 0.5, Last X index of 1.0.-. It really looks like the background circles around and the ball

reflects the portion that is "behind" the viewer. When it moves during an animation the effect can be stunning.

Mapping Types

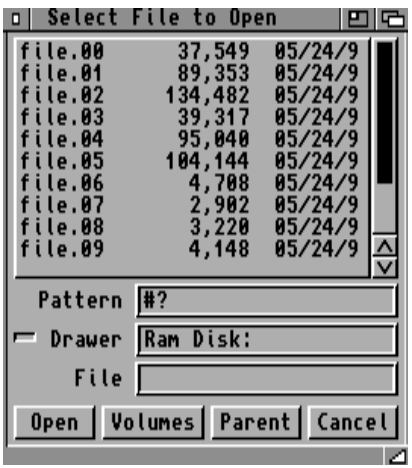
We are almost done with the texture tutorials, but we need to take a look at just what the mapping types do. When you first create a Texture List, its member is given the default map type of Projection along the Y axis. If you visualize the 3D space as it is oriented when you flat view in the Y axis, then think of your bitmap with its "up" pointed toward the negative Z and its "left" pointed toward the negative X, you can see how the Projection along the Y axis orients bitmaps. When you apply the texture, Aladdin 4D measures the selected polygons in the XZ plane, giving them a value representing their relative position when using bitmap textures. (Procedural textures actually use the real positions in space at the time of the assignment.) Then when you render, this value is used as a position in the bitmap. This even allows you to use a bitmap of a different resolution and color base without reassigning the texture.

Other projection, or wrapping, types are cylindrical, spherical, spherical point, Free Angle, etc. Click through them with the Projection Type gadget. You can wrap any of the three axes, selecting the one you want with the Axis gadget to the right of the Projection gadget. Again, the program will measure the polygons before applying the map if you're using bitmaps. If you are using procedurals, however, you specify the spherical wrap radius in defining the texture. You also can do this if you are using a bitmap with the TILES option ON.

Bitmap Memory Concerns

When you load a bitmap, it resides in memory until you quit the program or free the bitmap. It is loaded into the Texture TXResources area. This is the ONLY place it exists. You can refer to it and use it as many times as you like in the Texture Lists members, with virtually no additional memory required. All bitmaps are held in "fast" ram, not "chip" or "graphics" memory - so the limit to the number of bitmaps you can load is limited only by the amount of memory you have on your system. Amigas with one meg of chip ram are quite adequate for all tasks in Aladdin 4D as long as there is plenty of fast ram available.

When you load a bitmap that has relatives - that is, additional bitmaps with the same base name and a numbered extension like "file.00`1", Aladdin 4D considers this a sequence of single-framed bitmaps. The only the frame in current use is held in memory, so the length of the sequence is determined by your hard drive size, not memory.



Tutorial Conclusion

In these tutorials you have experienced quite a lot about texturing in Aladdin 4D. There are many more options you can use which are variations on the methods you have experienced. Experiment and play with them all, either as a few days of play, or as you need them in your work.

CSpline Tutorial (Cspline)

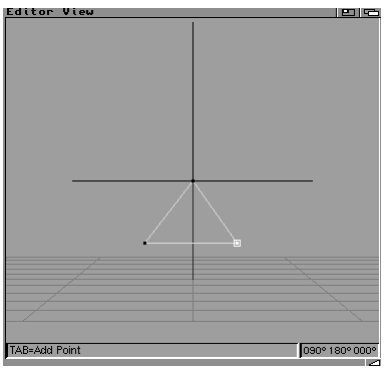
Aladdin 4D is, at heart, a program for animation. Almost every feature in the program has entry and exit values across a span of time. You can change color, transparency, rotation angles, scaling, deforms, texture strength and color level, and on and on.

To get what your imagination can envision, however, you need this control over these elements, and Aladdin 4D's CSplines and their Editor are here for just this purpose. You can produce fine animations without ever touching a CSpline, but you'll be limited to linear effects, and you might end up doing much more work than you'd do if you took the time to master CSpline control. CSplines remove linear limitations, applying the power of mathematical curves to your motions, morphs, whatever, for a full range of non-linear changes.

CSpline stands for "Control Spline." A Spline is a curve you draw in the Editor, and a CSpline controls things in a mathematically calculatable manner, like a Spline - not square and linear, but constantly in a state of change, or curvature. Occasionally, or for shorthand convenience, CSplines might also be called Splines and vice versa. Though they're the same concept, they're different in function.

First look at CSplines

Let's experiment with CSplines using a simple animation. Select New and Flat View in the Y axis. Use the Freehand Poly tool to draw a straight line from the origin, up the screen. Make the line about 20000 units long (the length of the Z axis in the Editor is 20000). Set the Attach Point to the bottom of the line (at the origin) by clicking on this point and using the Set command (RMB). Use the Freehand Poly tool to draw a small triangle from this point. You should have something like the illustration.



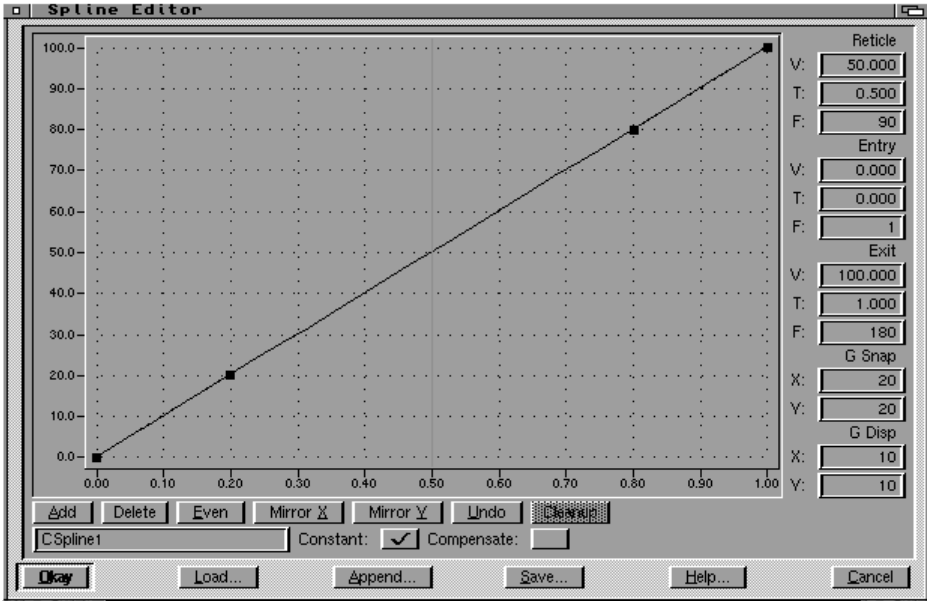
Now select the straight line you've drawn and choose the Path menu item Make Path. Turn ON only Movement Status. Do not change any of the other gadgets. Accept the requester. Select the triangle and use the Path menu item Assign Path. When Aladdin 4D prompts you to choose a path, select the straight line. The program Sets the ~~polys~~polygons to show you that the operation was successful. Now use the Render menu item Preview Animation. Change the number of frames to 180 and range from 0 to 180. Accept the requester. You will see the triangle running up the path, then repeating. This is the basic animation with which we'll explore CSplines.

How does the program know where to place the poly during each frame? ? It's figuring out how much of the animation is complete, then moving the triangle to a corresponding point on the path. For instance, when it paints frame 90, it knows that 50 percent of the animation has completed and places the triangle 50 percent of the way along the path. At 10 percent of the animation, the triangle is 10 percent of the way along the path, and at 90 percent it is 90 percent of the way across the line. This is a linear response, and if rendered to animation
the speed of the poly along the path would be instantaneously accelerated to a constant speed that it would maintain throughout the animation,, at the end of which it would instantaneously stop.

Wouldn't it be nicer to have the triangle start slowly, then speed up through the middle of the path, and slow down just before coming to rest at the finish? ? The answer is: yes, it would. Nicer, as well as more

natural looking. Objects in the real world don't accelerate instantaneously - no matter what the sales person claims - and unless they hit something substantial, they don't come to an instant stop, either.

Let's set up a CSpline for the triangle's path that models the real world better, and create a more natural-looking motion, acceleration, and braking.



Return to the Editor (<Esc> key). Select the path if it's not still selected, and choose the Path menu item Edit Path. In the movement area, which controls translation, click the CSpline gadget. The CSpline Selection requester opens. There are no CSplines to choose at this time, so click the New gadget to add one. This brings up the CSpline Editor. You can read about it extensively in the references section.

The new CSpline you added is now shown. It begins life as a straight line, which you can edit to your liking. The spline itself is bottom to top, left to right of the big window in the middle of the requester.

Along the bottom of the are divisions of time corresponding to the grid display. This area always reads 0.00 to 1.00. Along the left side of the editing requester are a range of values. These will be different, depending on what the CSpline you're editing is in charge of controlling.

For translation, and many others, it reads from 0.0 to 100.0, as a percentage of completion. You can change the display by editing the G Display gadgets at the bottom right of the spline. Try this. Large numbers will overlap the printed values and make them unreadable. You will only want larger numbers when zoomed in on a specific area and need a finer grid.

The red vertical line is the reticle. It can be moved to check the percentage values along the CSpline at any particular time. The numerical gadgets in the lower right display the current settings. The one marked Reticle shows V (value) is 50. That means 50 percent of the way through the animation. The Frame gadget shows which frame the reticle points to. In this case, 90. This is because we have 180 frames specified.

Click the reticle anywhere there is not a CSpline point and while holding down the left mouse button, drag left to right - and watch the numbers in the gadgets. Their values change with the reticle's position. You can use the reticle to see exactly what value will be created by the CSpline at exactly what time and frame number.

This new CSpline has only one segment. It can have as many as you wish, but we'll use just one for this first example. Like the Splines you draw in the Editor window to define curved polygons, this CSpline has a first point, two control points, and an end point. The first point is at percentage 0.0 and time 0.0. The last point is at a percentage of 100.0 and time 1.0. The control points are about 0.2 and about 0.8 in time, but this is not important just yet.

Control points define only the slope or curvature of the CSpline. The main points control the percentage values.

Select the first control point (click it with the left mouse button, and hold the button down) and drag it to a time of 0.5 and a percentage value of 0.0. When you select the control point, it turns red and the spline turns white to show you it is selected. When you have the control point where you want it, let go of the mouse button, move the mouse pointer to a place in the requester that has no points and click it. The spline will turn back to black and unselect. The control points are connected to the main point on the spline by a straight line. This shows you which point they control the slope for. The spline is now a curve. Now select the second control point and move it to time 0.5 and percentage value 1.0. Unselect the spline just as you did when you moved the first control point.

Look at the spline and think about what values will be created now, as time goes by during the animation. This is no longer linear. Select the reticle and move it to time 0.0. Now move it slowly at a constant rate until it is up to 1.0. Watch the percentage values reported in the Reticle Value gadget. The value changes slowly at first, then quickly, then slows down again as you approach the 1.0 value. Lets see how it affects the triangle's motion.

Accept the CSpline Editor requester, the Selection requester and the Path requester. Press <F9> to enter Preview again and watch what happens. The triangle is now translating along the path, but the CSpline is controlling position. It "eases in" to the motion and then "eases out", as graphically specified when you drew the CSpline.

Motions in nature behave this way, even manmade motions like an automobile's acceleration along a race course. Just as in nature there is no instantaneous acceleration or braking, your animations which conform to the same rules. How about a bouncing ball along a perfectly vertical path? It comes to a complete stop at the top of its path, then accelerates with gravitational pull on its way down. When it bounces, it goes (albeit quickly) to zero speed in one direction, and then back to bouncing speed upward. It's doubtful there is a motion that you can't create using Aladdin 4D's CSplines.

Return to the Editor. Select the path and choose the Path Menu item Edit Path. When the requester opens, click the CSpline gadget in the Motion area. The CSpline selection requester opens and the CSpline you created is highlighted. Select the Edit gadget, and the CSpline Editor opens again.

Bouncing the tTriangle

We'll make the triangle behave like a bouncing ball. The triangle is initially at the bottom of the path. If it were a ball bouncing, this would be where the ball has just hit the ground. The ball would quickly move away from the ground, slowing at the top of the path, hesitating a little, then slowly begin descent, accelerating until it hits the ground, and repeat. We could imitate this motion with a single CSpline segment, but it'll be easier to do it with two. Click the Add gadget. It's at the bottom of the Spline Editor requester. The program makes a second CSpline, showing it in the same window.

Also click the Even gadget, so the two segments will each get half of the time available. Select the end point of the second segment and move it down to a 0.0 percentage (at time 1.0). Move its control point to a time of 1.0 and a percentage of about 20.0. Move the control point for the first point of the first segment to a time of 0.0 and a percentage of 20.0 on the other side, to match.

The center point, where the two segments join, has two control points, one on each side. Select either one of them. We want the slope to be the same on each side. There is a shortcut to do this. Hold down the <Ctrl> key while moving one of the control points. The other control point takes up an corresponding position and moves in a coordinated manner. Move the control point to a percentage of 100.0 and a position as shown in the illustration.

This paraboloidal CSpline will cause a realistic bouncing motion. Accept all requesters and press <F9> to see the effect.

What if you wanted the triangle to sit still at the bottom for a while before bouncing? Go back to the CSpline Editor. Click Constant to turn it off. You can now move the ends of the CSpline away from the sides, and/or away from each other. Select the first point of the first segment (now at 0.0 time) and move it to 0.2 of time and still at 0.0 percentage. What do you think will happen now? Find out by previewing. The triangle stays at the bottom until the animation is 20 percent (0.2 of time) finished, then performs the bounce.

When you turn off Constant, you can move the segments of the spline away from each other. So what happens in the "in between" areas that are not covered by a CSpline? The program maintains the last value from the last segment until it hits another segment. To illustrate this, return to the CSpline Editor. Set up a CSpline similar to that shown by adding segments and positioning them.

CSplines have other interesting options, described in the section on the CSpline Editor. Compensate, at the bottom section of the CSpline Editor requester, lets you compress or expand the distribution of a spline's points, based on the relative position of its Endpoint as you move the mouse. Append opens a requester where you can set up defaults for appending a spline that has been previously saved to disk. That is, you can add CSplines together to build a more complex function. Aladdin 4D's Spline Editor allows non-constant splines - where X axis OR Y axis differs.

Tutorial: (Shadows)

This section discusses and gives examples of using shadows in Aladdin 4D. Shadows are an enigma in the world of 3D graphics. They require considerable time to calculate. However, adding them results not only in a more "natural" looking scene - at the price of much longer rendering time. Most 3D artists use shadows judiciously when required to achieve the effects they want.

Important: Shadows are different from shading. Shading is the engine of 3D graphics. Cast (or other) shadows are the troublesome items of which we speak here.

Shading a sphere, for example, simulates a shaded area on the dark side of the sphere without using shadows but by the calculations of the rendering process itself. The only time you need shadows is when you want the polygons to prevent light from striking a more distant polygon that is usually part of another object. An example is a sphere suspended over a plane - and you want the shadow of the sphere to show on the plane.

Much of the time, shadows are not needed - shading takes care of fooling the eye quite nicely. Some 3D artists use partially transparent polygons to simulate shadows with less rendering time than real shadows entail. These polygons can be calculated much faster than shadows, and sometimes can actually be more convincing, since you have control over their color and transparency levels. You can even texture them. Shadows are generally avoided when rendering animations because of longer rendering times.

With these cautions in mind, however, shadows in Aladdin 4D can be quite impressive. Aladdin 4D provides several methods of optimizing them for speed and realism. Let's experience these by example.

Shadow Acceleration

/// THIS IS DIFFERENT in al031 0505 /j

Select New, and Flat View in the Y axis. Use the Edit menu item Make Rectangle. Make the rectangle with X, Y, Z dimension of 20000, 20000, 0 and with a Z center of 10000. This rectangle lays flat in the XY plane down below the origin. Now use the Edit menu item Make Rectangle again. Use X, Y, Z dimensions of 10000, 10000, 10000 with X, Y, Z centers all 0. This cube is centered to the origin, and suspended above the lower polygon. Set the polyspolygons. Move the Attach Point to the origin (RMB on Page Move gadget).

Select the Object menu item Light / New. The Light Source requester opens. Turn ON Cast Shadows and accept. You will see the light appear at the current Attach Point. It looks like.. well, like a light, with rays pointing out of it. Move the light upward and to the right until it is near the top of the screen. Visualizing the shadow of the cube on the plane from the light will help you decide where to place it. Set the light.

You have to let the polygons know whether they should receive or cast shadows. In Aladdin 4D, these are independently controllable for each object, each polygon, in a drawing. Select the cube. Tap the <a> key and make a new Attribute List with the New gadget i. In the Edit Attribute List requester and name this "cube". In the requester's Options panel, Turn ON Cast Shadows. Leave Receive Shadows and Self Shadows OFF. Accept the requesters and Set the cube. Select the plane. Make a new Attribute List for it, too, but this time turn Receive Shadows ON, and leave the rest of the Shadows gadgets OFF. Leave Samples

~~//~~ ~~@~~ (Shadow Acceleration, really) at its default of 3~~.~~. Shadow Acceleration only affects those ~~polys~~polygons that receive shadows.

~~//~~ ~~@~~ ~~IS SHADOW ACCELERATION CONTROLLED IN "SAMPLES" GADGET ??? /J~~

Accept the requesters and Set the plane~~.~~. Objects that cast shadows can have any type of shading, but objects will only receive shadows if they are Phong shaded~~.~~. Select the plane~~.~~. Select the Object menu item Shading~~.~~. Turn ON Phong and accept the requester.

You're ready to render~~.~~. In Render Settings, select Light, Fill, Phong and Shadows, and select a Render Screen of your choice~~.~~. Render the image~~.~~. The rendering process slows down when it hits the plane~~.~~. Actually, it slows down when it hits an area of the plane that is NOT in shadow~~.~~. Let's talk about how the shadow is found~~.~~. When the program comes to a polygon that is to receive shadows, it must first find the position in 3D space on the polygon that will represent that polygon's pixels on screen~~.~~. It then must look (from this position) in the direction of the light(s) that cast shadows and see if any polygons are between this position and the light~~.~~. If so, the area is shadowed, if not the area is light~~.~~. This seems simple enough (you're sheltered from the math!) so why do shadows take so long~~?~~ ?. Imagine that the program has just found the location on the plane~~.~~. Next it must check ALL polygons to see if they are in position to make a shadow.

These tests take the most time~~.~~. The more polygons there are, the more tests must be made, and the longer it will take~~.~~. With a simple cube, there are only six polygons to test, so the render is relatively fast~~.~~. But imagine what kind of render time you'd get if there were a few hundred or even thousands of polygons!

This is what the Shadow Groups are for~~.~~. When you place a bunch of ~~polys~~polygons in a Shadow Group, the program will create bounding cubes for them~~.~~. When the program tests for shadows, it first tests the bounding cube, and if one of the sides of this cube would cause a shadow, then it tests all the polygons in the cube's Shadow Group to see which one (if any) cast the shadow - and then makes the appropriately shaped shadow~~.~~. This allows you to effectively control how many tests the program has to make~~.~~. Let's see how much faster it is.

Return to the Editor~~.~~. Select and delete the cube~~.~~. Use the ~~Prim q~~Primitive-Quad tool to make an ellipsoid with N1/N2 of 2.0/2.0; Radii of 10000, 10000, 10000, and Seg1 of 24 and Seg2 of 12~~.~~. A simple sphere appears~~.~~. Now you have the same scene as before, but the cube has been replaced by the sphere~~.~~. Select the sphere and tap the <a> key~~.~~. Choose the "cube" Attribute List and accept~~.~~. Render the image~~.~~. This render takes far longer than the cube (You can hit <Esc> to abort if you get tired of waiting)~~.~~. As you watch the render, notice that when the scanline being painted has a lot of shadow in it, it renders much faster - there is program optimization at work here~~.~~. The slowest scanlines are those that have little or no shadows.

Return to the Editor and select the sphere~~.~~. Click the right mouse button on the Group Level display gadget until it reads "Sdw"~~.~~. This is the Shadow Group~~.~~. Click the left mouse button on the Group gadget, to put the sphere in a Shadow Group~~.~~. Re-render the view~~.~~. This time the render is much faster, almost as fast as the simple cube~~.~~. The 144 polygons in the sphere have been surrounded by a bounding cube which is being tested first~~.~~. Then only if needed, the sphere's 144 ~~polys~~polygons are tested~~.~~. This happens, of course, for every pixel that the plane occupies on screen - a massive amount of calculation otherwise.

There may be situations in which you don't want to use Shadow Groups, like slats~~.~~. In general, you should use the Shadow Group if there are more than six polygons in the object and if they are not spread out in a very large area with large open spaces between them~~.~~. In this drawing, if you put them all in the same Shadow Group, the program will find a valid hit for the bounding cube (which surrounds all of the cubes) and will have to test all the cubes even in the middle of the plane, only to find that there wasn't a

shadow there after all. Imagine how bad this would be if these were all spheres. It would be better to use several shadow groups, so that the bounding cubes created were optimized for the drawing.

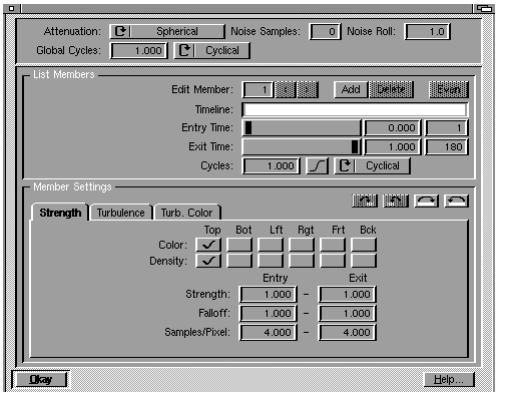
The same principle applies to complex objects. Say you have a machine with several moving parts. You may find it better to Shadow Group each part, or a few together, rather than to Shadow Group the entire machine.

Shadow Acceleration is a "lossy" technique. It only applies to those ~~polys~~polygons that are receiving shadows. Read about this in the Attribute List reference section (along with the other shadow gadgets). Remember that if you use too high a number, you'll miss details in the shadow. A good range is 1 to 5, depending on the object(s) casting shadows and whether they have "sharp" corners or protrusions.

In this tutorial, we didn't use an Attribute List for the light. This means it's using default values, so it is not as bright and not quite white. You may want to assign an Attribute List to the light to see the difference this makes. The global lights are also still at their default strengths, and dramatic differences can be seen if you turn them down, or off. It is normal to use at least some ambient light.

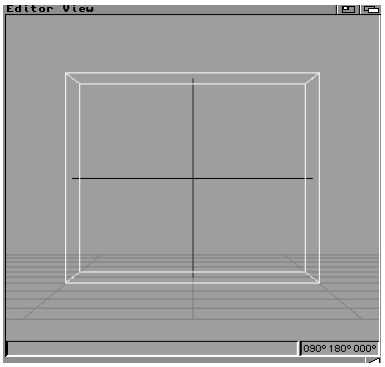
There's another very important aspect of shadows that you should be aware of. The mathematics used to determine whether a polygon is casting a shadow are very sensitive to whether the polygon is planar - in other words, flat. For polygons that are not flat, the shadows cast will not be correct. The easiest fix for this is to use only triangles for those polygons that exhibit problems. Triangles are always flat.

Tutorial: (Gases (Gases))



This section has tutorials on Aladdin 4D's gaseous objects. Gaseous objects are quite different from normal polygon objects. When you ask for a gas, you are asking to define a volume where you can modify the transparency of space. Unlike polygon objects, this is not surface oriented. Instead, the gas is treated as a solid. Indeed, the surface of the gas container is the part of the gas that is never used.

As the concept for rendering gases has evolved, it was decided that they should allow Attribute Lists and Texture Lists to control density changes inside the gas volume. We can define these lists to exist for polygons that form the container for the gas, defining its volume. This allows the user to see in the Editor a representation of where the gas exists, and be able to move it, resize it, and apply textures and Attribute Lists.



Select New and Flat View in the Y axis. Select the Object menu item Gas New. The Gas requester opens. Leave everything at defaults. Accept the requester and you will see a cube in space. This cube is the gas container.

Gas containers are regular polygons, but are treated quite differently by the program. You cannot delete only part of a gas container. It is treated as a whole. You cannot jump only part of a gas container into another space. If you move a point or a single polygon of a gas container, the program will make its best efforts to understand what you wanted to do, but the gas container will always be reconstructed. You cannot rotate a gas container except under path control during animation. In the Editor, they are always aligned with the axes. The specific parts of a gas container are named "Top," "Bottom," "Right," "Left," "Back", and "Front," and are referenced as though you have set Flat View in the Y axis.

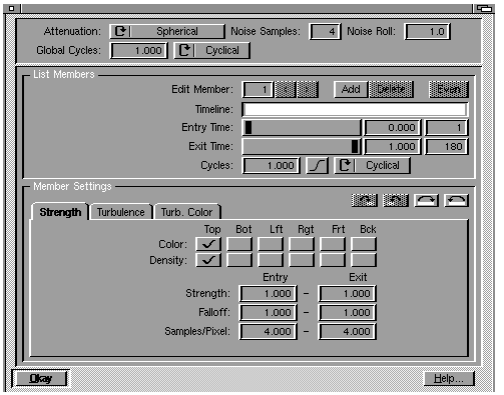
In order to see the gas in your renderings, you must do three things:

Apply Phong shading to the container.
Have some transparency in the Attribute List(s).
Have the Phong and Transparency global permission flags ON.

Of course, you must have the Fill option ON, not wireframe. If these conditions are not met, the gas containers will render as normal polygons.

Select the container and select the Object menu item Shading. Turn ON Phong and accept. The container is still selected, so select the Object menu item Attributes. Make a new list with the New gadget. Name the new list "Gas1". Get the Edit Texture List requester. Let's make this a bluish gas. Use color RGB values of 50, 100, 200. Set Transparency to 1. Accept the requesters. Select the Render menu item Render Settings. Turn ON: Light, Fill, Phong and Transparency. Set the screen mode to your liking. Accept the requesters and render the image.

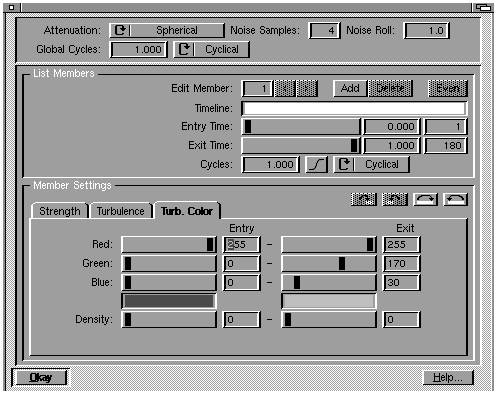
The image you see painting is the gas inside the container. After the image is complete, return to the Editor. Select the container and select the Object menu item Gas Edit. The requester opens.



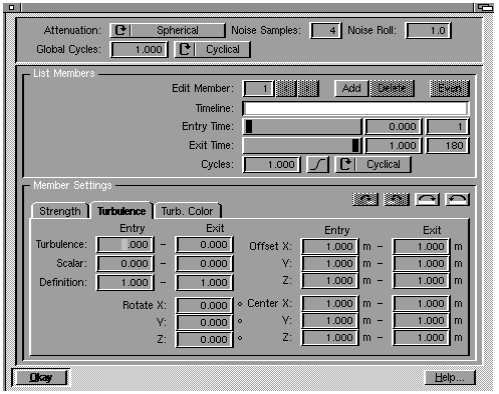
As is normal in Aladdin 4D, almost everything you can change for a gas has "entry" and "exit" settings to facilitate animation. A gas also can have an unlimited number of members in its list, time lined so that you can decide when changes occur. The topmost ~~rotary gadget~~cycle gadget, Attenuation, is set to Spherical by default, but this can also be solid, or specify different types of "falloff" or attenuation of the gas. Strength can be modified, as can the Falloff factor.

The "Samples/Pixel" gadget allows you to specify the number of gas samples to be taken by the program when creating the image. The larger this number, the more photographic the gas will be, but with a corresponding increase in rendering time which can be dramatic. Normal range for this is 1 to about 24. The program decides where to take its samples based on the attenuation type and whether there are any "solid" polygons inside the gas.

Let's look at Turbulence. The main control gadget is labeled Turbulence. Change it to 0.6. This means that the transparency of the gas is to be modified 60 percent by a fractal noise routine. Accept the requester and render the image. What you see is the gas, as before, but there are lacy streams of a different color running through it. These streams of different colored gas are the turbulence. The streams are not on the surface of the gas, but run inside it, like the marbling of color inside a cat's eye. During an animation this becomes apparent.



The color of the Turbulence is set in the Gas requester's Turb. Color tab, as well as its density. Its Definition and size are controlled by the Turbulence tab. Let's modify some of these settings. Return to the Editor and edit the gas. In the Turbulence tab, change Definition to 2.0 for both Entry and Exit. This will increase the "sharpness" of the turbulence.



Turbulence is controlled by the Turb. Color tab in the Gas requester. The usual specifiers for Red, Green, and Blue color are present. Currently, the turbulence is a yellowish color. Change it to pure red (255, 0, 0).

The Density sliders set the amount of density that you may want to add to the turbulent streams. Leave this at 0. Back in the Turbulence tab, change Scalar to 0.8. This larger number creates a finer pattern. A lower number creates large, sweeping turbulence. Let's also increase the amount of turbulence, by editing the Turbulence tab's Turbulence setting - say to 0.9

(both Entry and Exit).-._This will not change the size of the turbulence, but will lessen the strength of the gas (blue) around the turbulent streams.-._Accept the requester and render the image.

These settings cause a dramatic change in the appearance of the Gas.-._The increase in Turbulence actually decreases the amount of gas where there is not a turbulence stream.-._This has the effect of emphasizing the turbulence.-._The streams are made clearer not only because of the increased definition, but also because the density of the gas outside the streams (blue color) has been lowered.-._The scale of the turbulence is obviously finer (larger setting).-._Let's make it coarser.-._Return to the Editor and edit the gas.-._Change Scalar (in the Turbulence tab) to 0.2, and render the image.

This larger Scalar setting makes an obvious change.-._Notice, however, that the streams are not smooth in places.-._These areas are where the streams run closest to your line of sight - in and out of the screen.-._The low number of Samples per Pixel being taken (4) results in an unsmooth look.-._This is specified in the Strength tab.-._You can smooth the image by simply increasing the number of Samples per Pixel.-._Return to the Editor and set the number of samples to 8 or 12.-._More samples take much more render time.-._If you're not running a 68030, 68040, or 68060 Amiga, change your screen mode to a smaller image size (if you are in Ham, use the Custom setting, if using DCTV or a 24-bit board, just edit the number).-._This will give you faster feedback on the changes you're making.-._Render the image and you'll see a much smoother appearance.-._You can use a small number for Samples per Pixel during the layout phase of a drawing (sometimes as low as 1) to speed up the feedback, then go to a larger number of samples for the final rendering.

Return to the Editor and edit the Gas.-._At the top of the requester, Noise Samples is the number of times the fractal noise routines are reiterated to refine the turbulence.-._This is a much more subtle effect than increasing Samples per Pixel.-._The higher this number, the smoother the turbulence will appear and, of course, the longer the render will take.-._Normal range is 1 to 4.

Remember you can change the color of the gas using Attribute Lists.-._Let's do this.-._Notice the two rows of six gadgets labeled Color and Density.-._These are the surfaces of the container for which you can set the color and density.-._Currently both rows have Top selected.-._Select Bot (bottom) on both rows.-._The Top gadget stays selected.-._This is what we want.-._We'll use a different color for the bottom, so the gas will change color from top to bottom.

Change Turbulence to 0.7 so the container's color will be more obvious.-._Accept the requester.-._We want to change the color of the bottom of the container.-._The container is grouped, so hold down <Ctrl> and select one of the points on the bottom of the container.-._If one of the other sides of the cube selects instead of the bottom, use the Edit menu item Move to Last and try again.-._When you have selected the bottom only, tap
_the <a> key.-._The Attributes requester opens.-._the Gas1 Attribute List is selected because this is the list the bottom polygon is assigned to.-._Click the Clone gadget to duplicate the list.-._Edit it, and change the color to a bright yellow (255, 255, 0).-._Accept the requesters.-._Render the image.-._By the time you are a quarter of the way through the image, it should be quite apparent that the color of the gas (not the turbulence) is changing from the bluish color to the yellow.-._The sample that is taken has set its color based on its distance from the top and bottom (since these are the two sides being used).-._The closer to the top or bottom that the sample is, the closer to the color of the top or bottom the sample color will be.

You can change any of the object's sides to any color you want by the same process.-._Remember to specify the sides in the Gas Edit requester.

Gases can be resized.-._Return to the Editor and select the gas.-._Click the Resize tool and move the mouse pointer into the Editor window.-._Hold down the left mouse button and drag the pointer until the gas container is about half as tall.-._Render the image to see the result of the change in the container's size.-._The size of the container can be altered during animation, using Deform levels.-._This can be quite

useful for all types of effects.. Remember that the gas has a fixed amount of transparency.. As you make a container smaller in space (or look through a thinner section of gas) you will see the gas as a paler version of itself.. This is normal.. You can compensate for this if you want by setting the strength of the gas higher or lower.. Normal range is 0.5 and up.

We are almost finished with the tutorial on gases.. The last thing we want to show is the use of textures on the container and how the gas is affected.. There are two items of concern here: how the textures are oriented on the container, and what the various texture types do to the gas.

You assign a texture to the container's sides in the normal manner.. However, a procedural texture is treated completely differently than when used on any other object.. Since the gas is a 3D transparent solid, the procedural texture is actually indexed at the 3D coordinates of the samples being taken.

If you apply a bitmapped texture, however, the texture is applied to each side of the cube as though you were using a projection along the axis perpendicular to that side.. The bitmap on the top and bottom, for example, will appear in its natural position when viewed from the Flat Z view.. The sides will appear normal when viewed from the X, and the front/back will appear normal when viewed from the Y.. Refer again to the illustrations in the tutorial on textures.. The reason for this change in how the texture appears is that it allows considerable optimization for accessing the bitmap for the gases.. You can still, of course, use the flips and the index percentages to change the bitmap orientation.

Let's assign a texture to the container.. Select the Top of the container only.. Tap the <t> key.. The Texture List requester opens.. Make a new list using the New gadget.. Get the Edit Texture List requester and name it "gas_texture1".. Currently the requester displays the procedural textures.. If you haven't worked through the texture tutorial yet, this is all new to you.... so take a look at the texture tutorial if you need to.

Click on Bitmaps.. There are no bitmaps loaded yet, so select Load.. Load the bitmap image on the examples disk called "gaspattern.16".. then select it, and accept the requester.. Turn Color and Strength all the way up to 1.0.. Accept all requesters.. Open the Render Settings requester and turn ON textures.

Edit the gas.. Set the density so it uses Top only.. Set the color so it uses Bot only.. This means the density will come from the top, but the color will be the bright yellow we have on the bottom.. Render the image.

As you can see, the density of the gas is being "sculpted" by the bitmap.. The bitmap is a set of concentric rings of 16 greys, progressively lighter toward the center.. The program sculpts the gas based on the luma (highest value of RGB) in the part of the bitmap closest to the sample being taken.. This means that the gas will "descend" into the middle of the gas based on the brightness of the closest part of the bitmap.. That is why you see the cone shaped area cut out of the gas.. A Luma of 0, (black) will stop the gas at the surface of the container that is mapped.. A luma of 255 (for example, 255, 0, 0 or 0, 255, 122 or 0, 10, 255 or 255, 255, 0 or 255, 145, 255, etc.) will allow the gas density to extend all the way to the other side of the container.. A luma of 128 will extend the gas half way through the container.

If you wanted a galaxy with a cone shaped bulge in the center and spiral arms, you would make the bitmap that represented the depth, then use two gases, one on top of the other, with the bitmap on the bottom of the top one, and the top of the bottom one.. The results are quite remarkable - try it!

Animated bitmap sequences can also be applied to the faces of gases.

Using textures on gases in general requires more samples per pixel than just a straight gas, especially if the texture has hard lines in it.. A good trick is to use an extremely softened image, passing it through a blur

convolve a few times before using it as a texture. This usually achieves acceptable results even at low samples per pixel, shortening the render times.

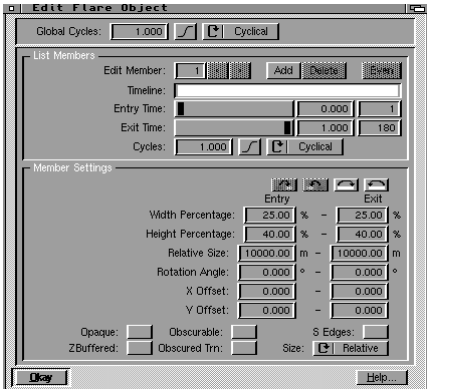
As you animate, you might be tempted to move the camera into a gaseous space. Aladdin 4D's extended gas render routines allow the camera to enter gas objects. However, when the camera goes into a gas container, the entire screen will be displaying a gas, so the render times increase significantly. Nevertheless, the effect is often well worth the extra render time, as the look can be extremely pleasing.

This basic tutorial on gases should get you inspired. Gases can be enlisted for almost any use - gas shrouded planets, galaxies, fog, solar flares, candle flames, photon torpedoes, rocket flares.. use your imagination. There's much more information there on the turbulence animation abilities specified in the turbulence modifier gadgets.

Tutorial (: Flares (Flares))

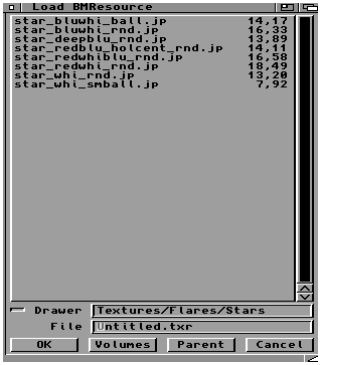
This tutorial will give you an introduction and general knowledge about the Flares in Aladdin 4D. These are multi-purpose objects not tied directly to lights, so you can use them not only to imitate lens flares (the artifacts that most photographers try to eliminate caused by bright lights reflecting in a camera lens), but also for simple fog, gas, light shaft, and image transition effects.

First look



Start the program, or choose "new" if you've been working already. Flat View in the Y axis. Choose the Object / Flare menu item New. The Edit Flare Object requester opens. We will change some of the defaults later. For now, just accept the requester. You will see a simple triangle appear at the current Attach Point (the origin). This triangle is your "handle" for the Flare. It is how you move the Flare and assign its attributes, textures, and paths. It can be changed in size and its points can be moved.

Select the Flare and choose the Object menu item Attributes. Add a new Attribute List, and name it "Flare_test". Give it a color of full green (R=0, G=255, B=0). The Color and Show flags are the only aspects of the list used by the Flare. Accept the requesters.



With the Flare selected, choose the Object menu item Textures. Add a new Texture List and name it "Flare_test". Change the type of texture to Bitmap. Click the Load gadget and load one of the "star" flares you find there.

After the texture loads, select it, and accept the requester. Move the Color and Strength gadgets to 1.0 and Accept the requesters.

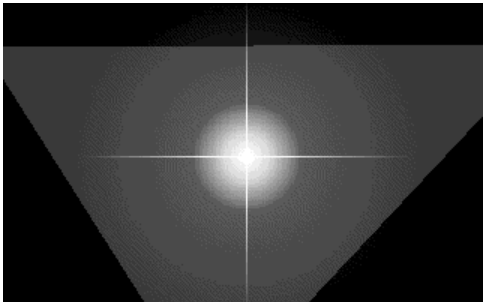
We want to see something besides the Flare, so let's just draw a simple triangle. Set the Attach Point at the top point of the Flare by selecting it there, and then Setting the object. Click the Freehand Poly tool and draw a line up to the left. Tap the <TAB> key. Draw the new line to the right to make a simple triangle.

Set the poly. Select the triangle again and move it up (hold the <Alt> key) a little away from the Flare, so it's not touching, and Set it. Select the triangle again, and as you did with the Flare, give it an Attribute List and color of medium blue (RGB 50, 50, 150 will do nicely.

The drawing is ready to render. Choose the Render menu item Render Settings. Turn ON Light and Fill. They are required to render Flares. Select the "set screen mode" gadget. Accept the requesters.

Render the picture. Watch as the picture draws. The program is actually rendering any polygons that you may have (just one triangle this time)

and then compositing the Flare image with it. You will see the Flare in the center of the screen with the triangle behind it. The "handle" for the Flare does not render, just the Flare image and the regular triangle you drew.



The image you see shows the Flare overlaying the triangle and the open space. The Flares use a type of transparency mapping when they are composited. The bitmap used for the image is mostly black. Where it is black the Flare image is completely transparent. As the "luma", or highest color component of the bitmap increases, the Flare gains density.

Resizing flares

You can change the size of a Flare, and rotate it. Let's do this. Return to the Editor. Select the Flare's handle and choose Object / Flare / Edit. The Edit Flare requester opens.

There are three ways to control the size of a Flare. The first two use the Size gadget. The third is through the bitmap indices in the Texture List. The gadgets that control the size of a Flare are Size (Relative or Fixed), and the Width and Height Percent gadgets If the Size gadget is set to "fixed", the Flare image size is the Width/Height Percents you specify - that means percents of the render screen size. By default, the Size gadget is set to Relative. In this case, the program decides how large a circle with a radius of the length you specify would be on screen, and sets the Flare's image size to this. This means with the size set to
relative, the Flares will get smaller as they get farther from the camera. The Width/Height Percents are then used to cause the Flare's image to be reduced or enlarged from this base size.

Let's see a fixed size. Click the Size ~~rotary gadget~~cycle gadget to change it to "Fixed". Accept the requester and render. The Flare image is a
direct percentage of the render screen size. The Width/Height Percents of the Flare are set to 0.25/0.4, which creates a "square" image on screen. Return to the Editor and change the Flare's Height Percentage to 1.0 on both entry and exit. Of course, you could use different values, and have the Flare change its Y size over time. Render this. First notice that it takes a little longer to initialize the Flare because it is much larger. The time required to initialize Flares depends on the size of Flare images, the number of Flares, and the number of Members in the Texture List the Flares use. This time, the Flare renders at 100 percent of the screen size in the Y dimension.

Offsetting

The Flare image is normally centered to the First Point of the Flare handle. You can offset this in the X and Y directions. Return to the Editor and edit the Flare. The OFFSET X/Y gadgets are by default at 0.0, centering the Flare image. Change the X offset Entry/Exit values to 1.0. Accept and render. The Flare image is to the right of its original position by the size of the Flare's width. Try a few different settings in both X and Y offsets. When you are finished, reset them to 0.0 for the next part.

Rotating

Let's rotate the Flare. Return to the Editor and edit the Flare again. Change the Rotation Exit gadget to 360. Leave the Rotation Entry gadget at 0. Accept the changes and render frame 1. This is the exact

image you just saw. Render a frame about a fourth of the way through. (If you have 60 set, draw frame 15.) At this point, the Flare will be rotated not quite 90 degrees (it would be on frame 16 of 60). Render a few other frames to see other rotations of the Flare. If you use offsets, the Flare still rotates around its first point allowing you to create some very fast light shaft effects.

Compositing

You can composite Flares in two ways. First, you can add another Flare and set its first point at the same position in space as the first Flare. (If you do this, move one of the points of one of the Flares so you can tell them apart. Only the first point has to be in the same position.) Second, you can add a second member to the Texture List. Let's do the second.

Select the Flare and choose Object / Textures. The Edit Flare requester opens with the Flare's texture selected. Add a member. The Member Number gadget changes to "2". The first member was cloned to get the second one, which you can edit as you choose. Load the "red_gas_ball" Flare from disk. Leave Color and Strength at 1.0. We want this member smaller and centered to the Flare. Set First X Index and First Y Index to -1.0 for both Entry and Exit. Set Last X Index and Last Y Index to 2.0 for both Entry and Exit. This will make the Flare occupy a smaller area, still centered. (You can read more about this type of change in the texture tutorials and the texture reference sections.)

Accept the requesters and render. Compositing the Flare images in this manner gives an unlimited number of possible Flare effects. Notice that you can "fade" one Flare image into another over time in this manner, through the Texture List's Strength settings. You can also change the size of the Flare's component composite images in the Texture List over time, along with changing the Flare's image size in the Edit Flare requester.

Flare's base color

We had you give the Flare a base color of green. Why? ? The Flare uses bitmaps from the Texture Lists. So what happens when the texture is not at full strength? ? The Flare gets its color as a composite of its Attribute List and the Texture. Let's take a better look at this.

Return to the Editor and select the Flare. Choose the Object menu item Texture. The first texture level has the "star" type image. Its color is set to 1.0. Change it to 0.0. Accept the requesters and render the Flare. The Flare image is now the base color as specified in its Attribute List. That green is fairly ugly, but great for demonstration. The Strength of the Flare is still inherited at 1.0, but you can change this, too. It is quite often used at levels lower than 1.0 if you want the Flare not quite so dominant, and especially if you have two or more members in the Texture List occur at the same time.

Obscuring Flares

Now you can change the Flares' size(s), composite their images, offset them from their centers, rotate them, and change their color and density. If this isn't enough, you can also have normal polygons obscure the Flares as they move through space (Gases never obscure Flares.) This means flares can appear to go behind some objects and in front of others, if you choose.

Let's try this. Return to the Editor. Put the Attach Point at the first point of the Flare (at the origin). Use the Freehand Poly tool to draw a straight line to the right, not quite off the screen. Set the new poly.

. We will make this as a path to move the Flare. Select the straight line by its rightmost point. Choose Path / Make Path. Turn ON Movement Status. Also turn ON Last - it's beside the Status gadget. This tells the path to move any assigned polygons along it, and back. Accept the requester. Now select the Flare and choose the Path menu item Assign Path. The program prompts you to choose a path. Click the rightmost point in the path. During an animation the Flare will obey the movement instructions of the path. Preview this to make sure it's all set up correctly.

Select the Render menu item Preview Anim. In the requester that opens set Frames to 120 and the range from 0 to 120. Accept the requester and you should see a preview of the animation. In the preview, Flares are shown as triangles, just like in the Editor. You should see the Flare running to the right, then back. The normal triangle should stand still.

We need to be sure that the Flare will go "behind" the regular triangle. Set everything. Then select the regular triangle. Select the X Active Angle gadget and press <Spacebar> to Flat View in the X axis. In the Y Active we were looking from a position that is now to the right on the screen. Hold down the <Alt> key and move the triangle to the right and down, so that when we look back from the Y view, it will be in front of the Flare. Set the polyspolygons. Go to the Y view by selecting the Y Active Axis and pressing <Spacebar>. Move the regular triangle to a position about half way along the path.

Rotate your view using the number keys to check what you have. Insure that the Flare as it moves along the path will go behind the polygon. Now when the Flare starts to move it is not behind the poly, then it will go behind the poly and emerge from the other side of it. This way we can see what happens when it is "obscured" or goes behind another poly.

Check three different views of the drawing.

TOP: Flat Viewed in Y active. X, Y, Z view angles of 90, 180, 90 MIDDLE: view from the right. X, Y, Z view angles of about 97, 180, 85 BOTTOM: view from above. X, Y, Z view angles of about 342, 0, 164

Okay, Flat View in the Y axis again and Set the polyspolygons. Edit the Flare again. Turn ON Obscurable. Accept the requester. Render the animation in a small format to get a good idea of what's going on.

Set up the Render for the animation and save this one, if you like - it'll be instructive. Use 60 frames with a range of 0 to 60. Accept the requester. Watch what happens when the Flare goes behind the polygon. It completely disappears. Watch what happens as it approaches the triangle.. it overlays the triangle. This is just what happens with a camera lens flare in real life as an intervening object intrudes on the source of the lens flare. In real life, of course, the flare has some "area". Aladdin 4D's Flares are point sources, and have no area, so the change is more abrupt. You could, of course, use Strength to do a "quick fade", but remember this would be tied to the camera view angle and would not work when viewed from a different direction. Usually the abrupt change is quite acceptable.

There are two more options to obscuring a Flare to discuss. First, what should happen if the Flare moves behind a transparent object? ? Should the Flare dim or should it be unaffected? ? Aladdin 4D lets you decide. If you turn ON Obscured Trn, the Flare will dim as it moves behind transparent polyspolygons. If this is OFF, the Flare will remain unchanged when it moves behind transparent polyspolygons. You can use this to advantage if you have a Flare inside an object, and you always want to see the Flare, but you also have other objects that should obscure it, set the transparency level of the containing object to "1" and turn Obscured Trn OFF. Then the Flare will be visible inside the container, but obscured by passing non-transparent objects.

If you have Obscured Trn ON, the Flare will dim as it moves behind transparent ~~polys~~polygons~~.~~. The dimming is accelerated by a factor of 4 as compared to the normal transmission of regular ~~polys~~polygons by transparent objects to simulate this effect, which occurs in nature~~.~~. Flares will only be visible if the transparent object has a transparency greater than 192.

To see the Obscured Trn option in use, change the Attribute List for the regular poly to suitable levels of transparency, turn ON the Transparency flag in Render Settings, and render.

Transparency

Another question is, should the Flare image disappear when the center of the Flare is beyond the screen edge, but part of the image could still be seen?

A true lens flare disappears when the source of the Flare is no longer in the scene~~.~~. But in case you want to use the Flares for some other purpose, you can specify that the Flare should be visible even if the "source", or first point of the Flare is off screen~~.~~. If SEdges is ON, the Flare will disappear when the Flare moves off screen abruptly~~.~~. If SEdges is OFF, the Flare image will continue to exist until the entire image is off screen.

To see SEdges option in use, just use the <Cursor Down> key to zoom in until the path's rightmost point is off screen.

Verify this in Preview - the render screen is larger than the Editor screen~~.~~. Then render some appropriate frame ranges in the animation.

Opaque flares

Flares and Fountains are frequently rendered as "additive" objects - their colors are added to the current color of the image at their location~~.~~. This is just what you want for fireworks, stars, camera lens flares, etc~~.~~. But what if you want to render them as confetti or transparencies?

Aladdin 4D's additive mode is the default~~.~~. But a gadget named "Opaque" lets you render them as transparent, instead.

If you turn ON the Opaque gadget, the Flare will be treated as a transparent object that you can use to create overlapping Flares which partly or fully obscure other Flares and ~~polys~~polygons in the drawing~~.~~. It also means you can create Flares as dark objects if you want.

You will find a gadget on the Edit Flare Object requester called ZBuffered~~.~~. Normal operation for a Flare is to suddenly disappear when a poly comes between it and the camera~~.~~. If this is on, the Flare image is treated not as a lens Flare source, but as a continuous source, so the occluding object can obscure only part of the Flare~~.~~. This allows you to place Flares behind objects, or even inside them.

If you want the Opaque option, you must use the ZBuffered option, too~~.~~. This is automatic~~.~~. If you select the Opaque gadget the ZBuffered gadget will be turned on for you~~.~~. When you ask for the Opaque option, Flares are treated as virtual polygons and added to the polygon base for the render instead of being rendered first and added at the end.

Often you will want a "cookie cut" for your Flare image when using the Opaque option~~.~~. There are two ways to achieve this~~.~~. First, you can set the Texture for the Flare to be Decal or Genlock~~.~~. This will use

"Color 0" of the image for a "no image" zone and will give the cookie cut effect. However, this effect will give you aliasing.

So.. to avoid this aliasing, you can use an Alpha Texture in the Texture List to modify the image of the Flare or Fountain. This results in Flare images that are of great quality regardless of their size. Let's say you want triangular shaped confetti. You would go into a program like ImageFX and draw the triangle shape on a black background, then save it. Then you'd create a Texture List and load the triangle bitmap. You then set the member's Type to Alpha, Strength to 1.0, and Color to 0.0. Hit the Match gadget. Turn ON the Anti-Alias gadget. Add another member. When you do this, it will duplicate the first member. Change the second one's Type to Color, and set its Strength and Color both to 1.0. Click the Match gadget again and accept the Texture List.

This list will give you the desired effect. Using Alpha type texture can give you much more subtle effects than a simple cookie cut. For instance, you can use it with the star images for Flares. Remember that dark colors are possible when in Opaque mode for Flares, so you may find the star images have darkened edges.

Conclusion

As you can see, Aladdin 4D's implementation of Flares is not limited to lens flare effects. You can probably find all kinds of uses, such as replacing simple spherical gases for faster rendering. They also make great misty, smoky, or underwater effects much easier and more realistic.

Tutorial: (Fountains)

This section has tutorials on Aladdin 4D's special Fountain objects. A thorough discussion of each gadget and its use is presented in the reference section on Fountains. This tutorial should get you going, and you can tackle the reference section for more specifics when you're ready.

Fountain objects are related to Flare objects, but with various extensions. Specifically, Fountains allow you to establish a specific number of Flare objects, their initial positioning and how they behave during animation. In addition, you can tell the Fountain objects how they should have behaved before an animation begins, using a feature called "PreRoll," a new concept in Fountains.

PreRoll lets you get a fountain "flowing" to your liking before you start your desired animation, without rendering frames to be discarded. In short, Aladdin 4D's is a fascinating implementation of a particle system that uses the Flare object technology to minimize memory usage and rendering times. Fountains can be used to create fireworks, star fields, magic wands, writing, trails, clouds, and so on.

Fountain rendering isn't for the timid, and certainly not for the impatient! Aladdin 4D includes an extensive implementation - capabilities within capabilities, all requiring thought and learning. So, be patient. There is much to learn.

Limits

Fountain objects, like gases, are not limited except for system memory and your patience. They render comparatively fast, if you resist your urges to ask for large numbers of particles and extremely short particle life spans, especially when PreRoll is on.

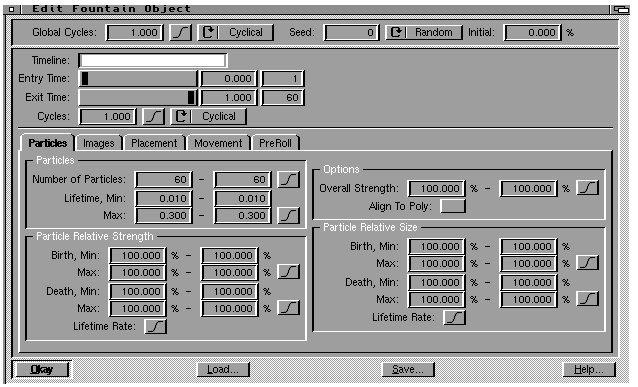
Be forewarned that there is no abort on Fountain initialization or rendering, so if you set up a Fountain that will take most of forever to render, you'll either have to wait it out until it's done, or reboot the computer. Understanding Fountains in advance will help you avoid this trouble. Basically, there are three things you can do to cause Fountains to take a long time to render.

1. Use a large image size for the particles (this can also happen if the camera is very close to a Fountain object)
2. Use a small range for particle life.
3. Use a large number of particles.

Further, if you render a specified frame of an animation, the Fountain must be calculated along the time line to the point in time you are asking for, since the particle positions are calculated from previous particle lifetimes and the particles' positions from their birth. This can take a while, especially if you've fudged on items 2 and 3 above.

As you gain experience with Fountains, you'll gain understanding of just how far you can push these parameters and still have tolerable rendering times.

First look



Start the program or select "new". Flat view in the Y axis. Choose the Object / Fountain menu item New.

As you can see from the resulting requester there are many aspects of Fountains to control. The requester's apparent complexity shows you just how much control you have. In addition, a few gadgets here open other requesters.

We'll change some of the defaults in later explorations. Just accept the requester. You'll see a simple straight line poly (two points) appear with one end at the current Attach Point (currently at the origin). This poly is the "stem" of the Fountain. As with Flares, it is your "handle" on the Fountain for assigning Textures and Attribute Lists, and for moving it around. The first point (point 0) of the stem, now at the origin, is the source of all particles generated by the Fountain.

Select the Fountain and choose the Object menu item Attributes. Add an Attribute List. Select Control and name it "fountain_test". Give it an Entry color of full yellow (RGB: 255, 255, 0) and an Exit color of full blue (RGB: 0, 0, 255).

Accept all requesters. Choose the Object menu item Textures. Add a Texture List. In the Edit window name it "fountain_test". Change its texture type to Bitmap and load one of the "star" textures. After the Texture loads, select it and accept the requester. Remember, you must select it.

Edit the texture and change its Color and Strength to 1.0. Click the Match gadget and accept the requesters.

Before you do anything else to change the Fountain's definitions, save this drawing as "fountain_tut.4d" (The file will be very small, about 3,500 bytes). You can reload it any time to get back to this point easily. These tutorials assume the Fountain exists, and they're fairly long, so you will probably not finish them at a single sitting. By saving the drawing to here, you'll be able to leave it at any time, then just reload it to continue.

Before we render the Fountain, it is important to realize that Fountains are dynamic systems. In other words, they change over time. Let's see this. Choose the Render menu item Preview Anim. Change the Number of frames to 180, and from 1 to 180. Accept the requester.

What you see in the Preview on the first frame is the Fountain stem. It is visible in the Preview and in the Editor, but it will not be visible in the fully rendered images. As the animation goes along, particles emerge from the Fountain stem's first point, the Fountain source. They emerge in a random fashion and move along a straight line away from the source. They live a while, then die. When one dies, a new one is born at the source to replace it. Remember all this occurs in 3D. Some particles are coming straight out toward you, and some are going away from you. The particles move inside a virtual sphere in space. After watching this so you understand how it all works, hit <Esc> to return to the Editor.

You probably want to see what a rendered Fountain looks like before we talk about controlling the particles' movement, so let's do that. We already have a texture on this one and an attribute list, so choose the Render menu item Render Settings. Turn ON Light and Fill. They are required to render Fountains. Set the render screen mode of your choice in the usual way. Fountains are objects of much

subtlety when rendered, so the best display mode you have available is the one you should use.. Accept the requesters.

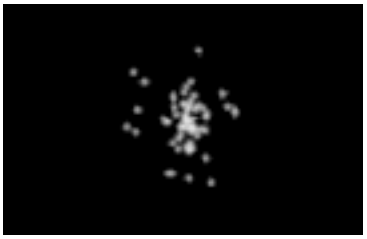
In order for Fountains to render, you must turn ON the Fountains toggle in Render Settings.. If you set a Fountain up, check its operation, and then don't want to wait for it to render every time you test something else, you can turn Fountains off in Render Settings.. Turn this on now.

Choose the Render menu item Render.. Remember the Preview?? Frame 1 has particles only at the source point.. In frame 1, you can expect to see particles only at this point.. So, another frame might make a more interesting demonstration.. Change the frame number to 60.

The render screen you chose opens with a wait pointer while the program initializes the Fountain and updates the Fountain to the frame you requested.. In other words, the program is going to build the fountain's 60th frame by building each of the others in turn, then rendering number 60.. It doesn't actually render each frame, just calculates the positions, lives, and other parameters of the particles.. You cannot abort the render during this phase.. With the default Fountain, this takes very little time.. however.. Remember, Fountains are "grown." The program must calculate where they've been in order to render where they are at any given point.. For large numbers of particles, this can take time.

The racetrack window opens to update you on the progress of the imaging of the Fountain.. This is an invisible operation - you can't see the image being formed, so the racetrack window keeps you posted on its progress.. You cannot abort the render during this phase, either.

Once the racetrack window closes, the final compositing will take place.. If you had any polygons in the drawing, they would be rendered first, with Fountains being composited into the final image.. You will see the Fountain particles being rendered with the star Flare image you chose.. Notice that the Fountain stem does not render.. Don't stop yet..., since the particles are in only a small area in the center of the screen, you won't see the particles being rendered till the scanline render reaches the area they're in.



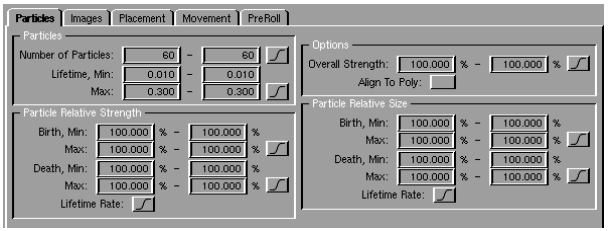
This is not a very impressive image.. If you call your family in to look at it, their yawn will be more impressive than the image.. Don't worry, we'll get to the impressive stuff later.. First, a basic understanding of the mechanics of rendering Fountains.

Fountains, like Flares, use a type of transparency mapping when they are composited.. The bitmap used for the image is mostly black.. Where it is black the particle is completely transparent.. As the "luma", or highest color component of the bitmap increases, the particle gains density.

Return to the Editor and let's begin a fairly in-depth look at what controls you will ordinarily use on Fountains.. In normal use you will want to control: the number of particles, Particle Relative Size and distance factors, Particle Movement and Placement, and perhaps in the PreRoll tab: PreRoll Enabled (of course) and Old Age Splines.. In the Images Tab: Attributes/Textures, you'll need Particle Image Size.

Of course, all of the controls are there, because at some time or another for some purpose or other, you might need them, so don't be surprised if you find yourself using something that isn't in this abbreviated

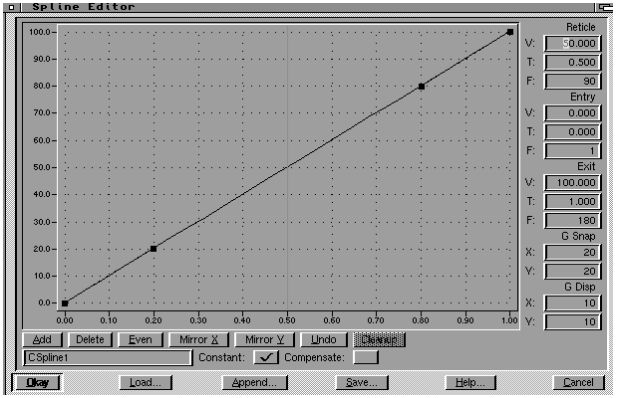
list. Notice, also, that many of the parameters you can set can be assigned to a CSpline for maximum control.



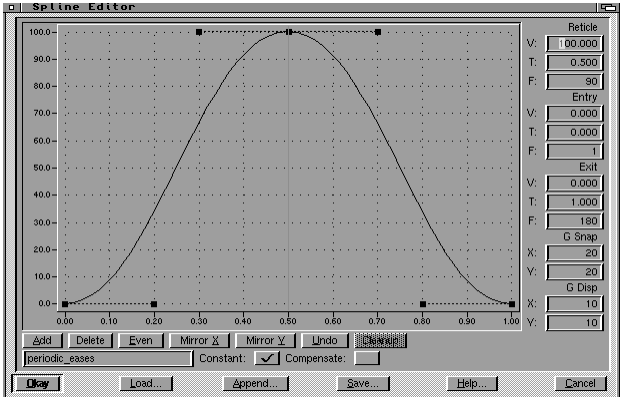
Fountain Number of Particles

Select the Fountain (only) and select Object / Fountain / Edit. The Edit Fountain Object requester opens. The Particles Tab in the requester contains a box of goodies for specifying the Number of Particles - as usual for each of Entry and Exit. Whatever numbers you enter here, you're telling the Fountain to maintain this count. As a particle expires, a new one is created to replace it, so the count always remains stable.

Change the Entry side to 1, and the Exit side to 200. Accept the requesters and Preview. Notice that at the beginning of the Fountain's animation, the Fountain generates only a few particles, and at the end, far more. The change in numbers of particles is linear. It can be made non-linear with the CSpline gadget beside the Number of Particles gadgets.



You can use CSplines to exert immense control over the number of particles animated in the Fountain - spurts, gushers, geysers, drips, anything you want, suddenly or gradually increasing or decreasing as you desire.

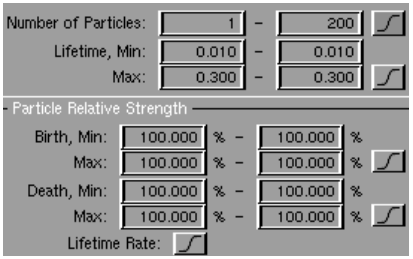


Click the CSpline gadget for Number of Particles. Add a new CSpline, and load the "periodic_eases.csp" file from your CSplines directory. If you have been using CSplines, you can easily see what this will do.

Time runs left to right. The Entry value is at the bottom and the Exit value at top. So, this CSpline will cause the Fountain to emit one particle at frame 1, and build to 200 in a nonlinear fashion described by the curve; then at the middle of the anim, the Fountain will begin reducing the count, again non-linearly, and end with the count at 1 again. Preview this to observe the effect.

Fountain Particle Life

As you watch the fountain animate, a particle is born at the source then moves to the outside of some random point on a virtual sphere. Only a few actually make it to the outside. This takes about 30 percent of the animation for them. The others never live long enough to get there. Let's change this.



Edit the Fountain. In the Particles tab, notice the gadgets named "Lifetime" They're there for Max and Min. The value of the Entry gadgets (on the left) is the "life" value the Fountain's particles will have if they're born at frame 1. The value of the Right gadgets is the value a particle born at the last frame of the animation will have. As the animation proceeds, the values are averaged so a particle born at 0.5 of the animation, or the middle frame, will get a lifetime halfway between the two values.

The two values allow you to set a range for particle Life. Max and Min are limits, not absolute values. A particle will get a random value between these limits. If you set a Max/Min of 0.3 and 0.1 at entry and exit, a particle will get a Life somewhere between 0.3 and 0.1 no matter when it is born. If you set different values for Exit, a particle's Life will change depending on when it is born.

Let's try a few changes. First lets make the number of particles constant, so it'll be easier to see the effect of particle Life. Change Entry/Exit Number of Particles back to 60 and 60. Click the CSpline for the Number of Particles, and click Free to remove it. This is a good habit to develop, to avoid confusion as you work.

Now for the Life values. To make this text easier to read, let's mention the values as a group of four values, Entry Max/Min and Exit Max/Min. So currently you see values of 0.3/0.01 0.3/0.01. Change these to 0.5/0.01 0.5/0.01 and preview (F9). The particles live longer just as you expected.

Edit the Fountain and change Life to 1.0/0.5 1.0/0.5 and preview again. Watch this one carefully. An initial burst occurs. The particles that are born have a Life between 0.5 and 1.0, and there are 60 of them. Then no new particles are born until one dies since the particle number is not changing. When the animation is half done, you see some of them die and at the same time, new ones are born to replace them and maintain the particle count. These new ones also have a Life between 0.5 and 1.0 - since there is only 0.5 of the animation left, none of these will ever die. As the animation goes beyond 0.5, you see other particles die and be replaced. These are the particles that were initially born with a Life between 0.5 and 1.0.

Edit the Fountain again and enter values of 0.1/0.1 0.1/0.1 and then preview the animation. The particles live 0.1 of the animation. With Max/Min limits the same, all particles will have the same Life values. At 0.1 through the animation, the initial 60 particles will all die at the same time and be replaced by 60 more, again all with a Life of 0.1. Is this what you expected? While you have Life set this way, change Number of Particles Entry/Exit to 0/60 and preview. Notice that all particles are reaching the edge of the virtual sphere and travel at the same "speed". This is a useful setting for many drawings.

Try different values for Life and preview the results. As you do, there are four main things to notice and remember:

1. A particle cannot be born until the number changes or an old one dies.
2. The distance a particle travels is related to its lifetime.
3. The total distance traveled is set by the longest life possible.

And VERY important:

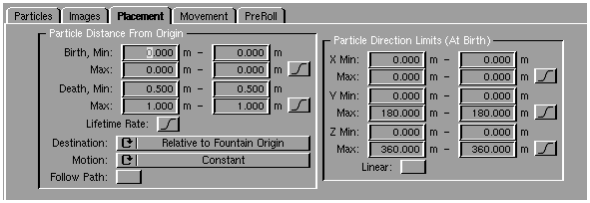
4. DO NOT use values of 0.0/0.0 0.0/0.0. These are not clamped.

So, if you do, you'll end up with an animation that shows no particles as expected. Also as expected, each frame will take longer and longer to calculate and could result in a crash of the system. Generally, do not use a value of less than 0.001, for this reason.

One more point of interest. The Particle Life values can range from 0.0 up. There is no upper limit. This is because if you need to use PreRoll, you may want values up to 2.0. Values above 2.0 are not useful at this time.

The smaller the range of Life you set for Max/Min, the longer the Fountain will take to calculate. Very small values, especially with large numbers of particles, can require very long periods of time to calculate. There is no abort during the calculations. Be careful in setting your Life values to very low values, especially if you're using a larger number of particles.

Particle direction and placement



The Fountain you have been experimenting with created a virtual sphere with its particles. Let's look at altering this. Edit the Fountain. Use Number of Particles of 60/60 with no CSpline. Set the Life values to their defaults of 0.3/0.01, 0.3/0.01, again with no CSpline. Click the Placement tab in the Edit Fountain Object requester. In the right panel, there are three directions, X, Y and Z each of which has Entry/Exit Max/Min values (and a gadget with which to invoke CSpline control). Let's take the Y direction for an example. Currently it has values of 180.0/0.0 180.0/0.0. This means a particle when it is born (whenever it is born, since these don't change) has a "Y direction" of between 180 and 0 degrees. So just what does this mean?

It is easier to see than to read, so let's look. Change the Y direction values to 180.0/0.0 0.0/0.0 and preview the animation. The direction of any particle is fixed at the time it's born. As the animation moves along, the general tendency of the Fountain is to move from a sphere to a single vertical line. This will be even easier to see if we get rid of the Z direction components. Change the Z direction values to 0.0/0.0 0.0/0.0 and preview again. You now only have the particles in a flat plane.

Edit the Fountain and enter Y values of 90.0/90/0 180.0/0.0 and preview again. This one should firm up your understanding. At frame 1, the particles have a direction of 90 degrees on the Y. All particles have this direction, so a straight line of particles is being emitted. As the animation progresses, the particles get a direction that is between two values that change up to 180.0/0.0 at the last frame. So, the straight line opens to a 180-degree fan shape.

There is a tremendous amount of control here, so experiment with the direction values to see what types of Fountains you can make. Some values of particular interest are - -given in Entry/Exit Max/Min on the X, Y, Z:

```
sphere:
X:   0.0/0.0    0.0/0.0
Y: 180.0/0.0  180.0/0.0
Z: 360.0/0.0  360.0/0.0

hemisphere:
X:   0.0/0.0    0.0/0.0
Y: 90.0 /0.0   90.0/0.0
Z: 360.0/0.0  360.0/0.0
```

```
quadrantsphere:
X:  0.0/0.0  0.0/0.0
Y: 90.0/0.0 90.0/0.0
Z: 90.0/0.0 90.0/0.0

disk in XY plane:
X:   0.0 /0.0    0.0 /0.0
Y:  90.0/90.0   90.0/90.0
Z: 360.0/ 0.0  360.0/ 0.0
```

You can achieve these in different combinations. For instance a sphere can also be achieved by using 360.0/0 on the Y values. It will be significantly different than the one listed and yet occupy the same area. Notice that if you use a value of 3600.0 instead of 360.0 for the Z direction you still get a sphere, but again, it is quite a different distribution. As you gain experience, you'll be able to visualize the results, and set the values you want rather intuitively.

An interesting note here is: What is the direction when values are all at 0.0?? This is the default direction.. If you try to Preview it, you will not see the particles.. This is because they are moving straight up.. That's where the stem is, so they are obscured by the stem in the Preview.. However, you would see them in the render.

Align to Poly

A related direction control is Align to Poly, which is a Boolean gadget in the Options panel of the Particles tab.. It is normally off.. Let's see what happens when it's ON.. Edit the Fountain to be a disk in the XY plane with the values shown in the table.

Flat View in the Y axis and Preview/ You'll see a straight line, because you're looking at the disk edge on, so rotate your view a little on the X axis by using the <7> key on the number pad, then the <8> key to stop the rotation.. Use a view angle of about 105, 180, 0.. This way you're looking slightly down on the disk.. Preview to make sure you have a disk.

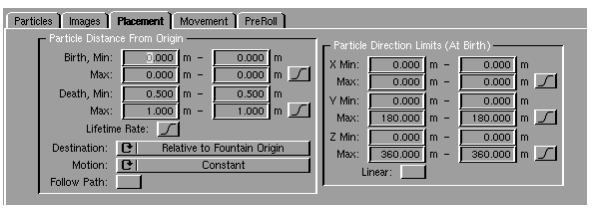
Return to the Editor.. Select the point at the top of the Fountain's stem.. Move it to the right slightly, so the stem makes about a 20-degree angle with vertical.. Preview this again.. There is no change in the disk.. Edit the Fountain and click the Align gadget (to turn Align ON) and preview again.. The disk is rotated about 20 degrees.. This feature isn't generally used, as you can achieve the same results by assigning the Fountain to a path, and then you can change the results over time.. But for some cases, particularly a single frame illustration, this method is easier.

Direction Linear

Another related direction control is the Linear check box of the Particle Direction Limits panel in the Placement tab.. If this is OFF - its default - the particles' direction is randomly chosen between the limits you set.. If it's ON, the particles are ordered by their creation between the limits you set.. For instance, if you have the Y direction set to Max/Min of 360.0, 0.0 and have 360 particles, with Linear off, the particles get a random Y direction between 360 and 0 (degrees) when they're born.. Particle 1 may get 247.6 and particle 2 may get 17.9, etc.. If Linear is ON, particle 1 would get 1 and particle 2 would get 2, etc.. With 180 particles, particle 1 would get 2 and particle 2 would get 4, etc.. This means you can get perfectly spaced circles of particles easily, using particles of the same Life at a fixed distance.

Later, under particle placement, we will explore using paths.. There, you will see some exquisite methods of control of particle position.. Other related controls are Drift, which we will look at later, and Distance, which we will look at now.

Particle distance



There are extensive controls over the distance from the source that the particle is created at and moves to.. Edit the Fountain again.. The Placement tab contains specifiers for Particle Distance from Origin as Min and Max limits for both Birth and Death.. Naturally, there's a CSpline gadget for these, too.

The values in the Entry column control particles' distance from the source when born, and the values in the Exit gadgets control the particles' distance from the source when they do the Commodore Business Machine (die).. We'll again list values for these as Max/Min, Max/Min.. When a Fountain is created,

these values are set to 0.0/0.0, 0.0/0.0, 10000.0/5000.0, 10000.0/5000.0~~.~~ This means all particles will be born at the source and have a destination between 5000 and 10000 units~~.~~ These entries have no limits - they may be any number, including negative values.

Let's change them a few times to get an idea of the power here.

First an expanding ring~~.~~ Change the values to 0.0/25000.0, 0.0/25000.0, and 0.0/0.0,0.0/0.0~~.~~ Preview this~~.~~ Notice that during the animation, the birth position of the particles expands outward~~.~~ The destination position is maintained at the source, so as the "ring" expands, the particles always head toward the source~~.~~ Most of the particles don't get there~~.~~ The default Fountain has the particles travel at a fixed "speed"~~.~~ If the longest lived particle is 0.3, and the largest distance to travel is 25000, only a particle that lives 0.3 will travel 25000~~.~~ A particle that lives half as long will travel only half as far.

You can toggle the Fountain so that all particles will travel their assigned distance, no matter how short they live~~.~~ In the Placement tab, the "Motion" ~~rotary gadget~~cycle gadget controls this~~.~~ By default it reads Constant~~.~~ Click it and it will change to Relative~~.~~ Preview this~~.~~ Notice that all the particles reach the source, with a resulting variation in "speed".

Another rotary selector gadget controls the type of Destination - that is, "Relative to Fountain Origin", or "Relative to Birth Place"~~.~~ If the former, the Destination values are distances from the Fountain's source~~.~~ If the latter, the values are absolute distances, measured from the particle's position when it is born~~.~~ Change it to each of these and preview, to get the concept of what it controls.

When Destination is Relative to Fountain Origin, the particles will all move toward a point located the specified distance from the source - to the source in this case, since the value is 0~~.~~ If you vary the strength of the particles so they are bright when first born and then fade off, this can be quite useful for modeling real world objects such as fireworks that start of bright and then fizzle out.

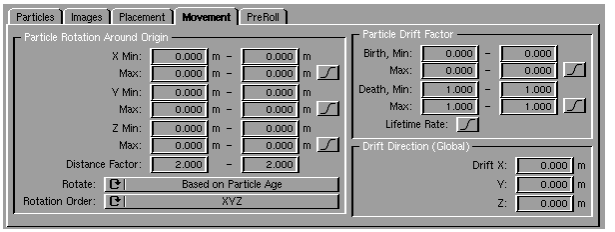
Another related option controls the global cycles of the fountain either pseudo-randomly, or in a pattern~~.~~ The control over this is the ~~rotary gadget~~cycle gadget in the upper right part of the requester~~.~~ If Random, its default, the program will try to insure that there are no recognizable patterns created by the particles~~.~~ If Pattern, the program allows recognizable patterns to be created~~.~~ The patterns are most obvious when creating a 2D disk type of Fountain, but can still be perceived with full 3D Fountains.

Edit the Fountain again and look at the CSpline gadgets in the Placement tab - mainly in the section called "Particle Distance From Origin." These CSpline gadgets permit maximum control of the rate of change between the values in the Entry/Exit boxes~~.~~ One CSpline controls Max/Min of each.

Notice ~~that~~ the CSpline gadget labeled "Lifetime Rate." ~~///check this 0420 /j~~ This is the Individual Particle CSpline gadget for distance~~.~~ If you look around the requester's other tabs you'll see this CSpline specifier for some of the other controls as well~~.~~ The Lifetime Rate CSpline gadget allows you to control the behavior of EACH particle during its life~~!~~! For distance, for example, if you load the Periodic_eases CSpline (or create a similar one of your own), each particle will move from its entry position to its exit position, then back again~~.~~ You will want to try many variations on the distance variables, with CSplines, pattern displays, varying the Number of Particles, etc~~.~~ When you find a Fountain setup that you like, save it for future use.

The first group of controls includes things you'll want to control every time you use a Fountain~~.~~ Next, we'll take up things you will want only occasionally.

Particle rotation



As you have discovered, you can change the direction a particle takes when it is born, and you can change the distance the particle has traveled when it dies. The Rotation tab allows you to change the "direction" of the particle over its life.

As with the Direction controls, there are three directions, X, Y, and Z, each with Entry/Exit Max/Min values. Let's take the Y direction for an example, again. Currently it has values of 0.0/0.0, 0.0/0.0. This means a particle will not rotate around the Y axis during its life. Let's change this and see the result.

You've probably been changing many values in the Fountain, so let's start with a new one. Don't use the "new" menu item. If you do, you have to load your texture and attributes lists again. Instead, create a new one in a new space. Set the drawing by clicking the right mouse button in the Editor window. Click the Space Control gadgets (bottom of tool box) with the right mouse button. The Space Control requester opens. Click Create. You have a new work space.

Flat View in the Y axis. Use the Object / Fountain menu sub-item New. The Edit Fountain Object requester opens with defaults in place. Change the direction values so the X and Z values are all 0.0. Change the Y direction values so they are a Max/Min of 360.0/0.0, both Entry and Exit. This will create a flat disk that is easily seen from a flat Y view. Preview this. (If you rotate around the Z axis in the Preview using the <1> and <3> keys, this becomes apparent.)

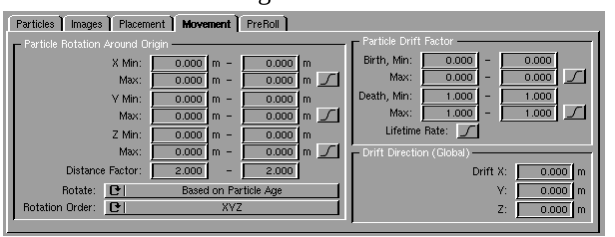
Edit the Fountain. Change the rotation Y values to 360.0/360.0, 360.0/360.0. Preview this. Notice that all the particles rotate 360 degrees around the Y. Change the values to 360.0/0.0, 360.0/0.0. (Remember, these are Max/Min values.) Preview this. The particles are now rotating by randomly assigned amounts between the Max/Min limits you've entered.

Change the values back to 360.0/360.0, 360.0/360.0. Click the Rotate button - It reads "Based on Particle Age" - to change it to "Based on Particle Distance". Preview this and you will see that the amount of rotation the particle gets is now related to its distance from the source. As a particle moves farther from the source, its rotation decreases. There is also a value for Distance Factor. It defaults to 2.0. Changing this to a lower number lessens the amount of decrease. A higher number increases the decrease. Try values ranging from 0.0 to 5.0 or so. Positive numbers give intuitive results. You can try negative numbers if you want, but the results are not very intuitive.

There is also an execution order gadget, to allow you to specify the order in which the X, Y, and Z rotations will occur.

Some results you can achieve with rotation are not at all apparent at first glance. Let's do a quick example to illustrate. Change the direction variables so they are all 0.0. As you have found out, this means the particles will all go straight up the Z axis - along the stem in this case. Use Y rotation values of 360.0/360.0, 360.0/360.0 and preview. Use the Periodic_eases CSpline you've loaded on the Lifetime Rate CSpline, and preview this with Rotation Based on Particle Age, then with Rotation Based on Distance. Some very interesting particle behaviors can be achieved using rotations.

Particle drift



You can specify a global "drift" for the Fountain. ~~All particles created by the Fountain will be affected by this~~This will affect all particles created by the Fountain. It can be used to simulate "wind" and "gravity"~~.~~ Let's try it.

Delete the Fountain you have in Space #2~~.~~ Add a new Fountain~~.~~ In the Edit Fountain Object requester, click the Drift tab at the bottom of the requester~~.~~ In the Drift Direction (Global) section, there are numeric entry boxes for each of X, Y and Z~~.~~ They default to 0~~.~~ Change the Z Drift to 20000 and Preview the anim~~.~~ You know that a Fountain at its default occupies a spherical space~~.~~ This is obviously different~~.~~ The particles are being moved to their position for the current frame, then the amount of Drift they have experienced by that frame is added to their position~~.~~ The result is what you see.

Edit the Fountain again and let's have a look at the gadgets in the Particle Drift Factor portion of the Drift tab~~.~~ Here again you see Orig Max/Min and Dest Max/Min as with Distance~~.~~ These values represent a multiplier of the amount of Drift you have specified~~.~~ In this case, 20000 on Z~~.~~ The default settings tell the program to apply 0 percent of the drift at the particles' birth and 1.0 (or 100 percent) of the drift at the particles' death~~.~~ You can use any combination of values for the multipliers, including negative numbers~~.~~ If you use different numbers for the Max/Min values, the program will distribute values between the limits you set, but at random for the particles as they are created.

Notice there are also three CSpline controls here~~.~~ Their use is identical to that you learned for Distance~~.~~ Click on the Individual Particle CSpline and use the Periodic_eases you have loaded~~.~~ Preview the anim~~.~~ The behavior of the particles directly reflects the CSpline's curve~~.~~ Edit the CSpline again and change the Lifetime Rate CSpline~~.~~ Since you are not using the CSpline that already exists, called CSpline1, select it~~.~~ Control it and move its left control point to the bottom, halfway across~~.~~ Move the right control point to the right, halfway down~~.~~ This is a "slow to fast" type of CSpline~~.~~ Accept the requesters and preview~~.~~ This makes a pretty good simulation of gravity's acceleration~~.~~ You can also use Drift to "decay" a pattern that was drawn by a Fountain under Path control - as you will find out in the next section.

Particle placement with paths

Paths offer a most interesting and exciting aspect of Aladdin 4D's Fountains~~.~~ You can move the Fountains with paths, and control whether the particles move along with the path, or are created at the point the path would be when they are born~~.~~ This means you can do magic wands, comets, or even write your name with a Fountain~~!~~! We'll illustrate with a few simple examples.

Use the Fountain you have been working with~~.~~ It should be nearly default, except Drift should be 20000 on the Z, the Particle Drift Factor values should be 0.0 in all the Orig gadgets and 1.0 in all the Dest gadgets~~.~~ Use a Lifetime Rate CSpline on the Drift factors - a "slow to fast" type~~.~~ Select the Edit menu item Make Arc~~.~~ Use 360 degrees, 12 segments, with one radius of 10000 units~~.~~ Select the circle and use the Path menu item Make Path~~.~~ Turn ON Movement Status and Last Segment~~.~~ Also make the exit rotation value on the Y axis 360 and use 3 cycles on the Y rotation~~.~~ Accept this~~.~~ Select the Fountain and assign it to the path~~.~~ Preview the anim~~.~~ As you can see, this is a perfect "magic wand" simulation~~.~~ The particles obey the path's instructions at the time of their birth, then ignore them from then on.

Edit the Fountain again~~.~~ The gadget labeled Follow Path in the Placement tab is currently not checked.~~///Move Type - /// Move Type does not exist in either Fountain or Path requester ??? 0421 /j~~ Select it~~@ reads Independent. Change it to @ read Grouped~~, and then preview~~.~~ The particles obey the Path @ instructions over their life~~.~~ This is useful if you want to move @ a Fountain along a path, keeping the particles' relationship @ to each other consistent.

Unselect the Follow Path checkbox.@ Turn ///Move Type back to Independent.

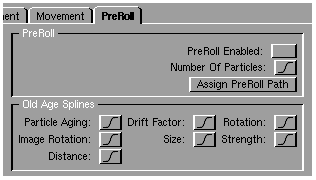
@ Set Number of Particles
@ to Entry/Exit values of 0/200.. Change the distance values so they
@ are all 0.0. PrPreview the anim to see the results.

Okay, get rid of the drift by changing Drift X, Y, Z all to 0.0.. Also free the CSpline you have in use for Drift - not really necessary, just good housekeeping.. Change the Life values to 1.0/1.0, 1.0/1.0.. Preview to see the results.

Since the particles live for the length of the animation, they never expire.. Since they are created evenly over the length of the animation, they are deposited along the path described by the Fountain's movement, making a kind of 3D spiragraph.

Infinite numbers of effects are possible with Path control over Fountain particle placement.. Go ahead and play with all types of paths and values for the Fountain..... but first, let's explore PreRoll, since it's directly related to the Fountain you now have in the drawing.

Fountain PreRoll



With the Fountain settings as in the previous example, you notice that the particles are created from start to finish along the path.. What if you wanted the pattern to exist when the animation starts?? That's what PreRoll is for.

Edit the Fountain.. Click the tab called PreRoll.. Th controls here let you have Aladdin 4D calculate the fountain to a given position before beginning to render the frames of your animation.

In the Edit Fountain Object requester, over in the PreRoll tab, there is a gadget labeled Assign PreRoll Path.. Click it.. The requester goes away and you're prompted to choose a path.. Select the circular path.. The Edit Fountain requester returns.. There are several other controls you can use here, but we won't deal with them now.. You can read about them individually in the reference section as your understanding of Fountains increases and you need to use them.

In the Particles tab, click the CSpline for Number of Particles.. Add a CSpline and click Control.. Adjust it so it is a straight line across the top of the CSpline Editor requester.. Accept the requester and preview.

This is what is happening..... It may appear to you that the pattern was made before the animation began, and is being sustained through the animation, but this is not entirely true.. Really, the PreRoll function runs the Fountain through a time range from 0.0 to 1.0 before the anim begins, getting the particles into the pattern you see.. Then as the animation begins, the particles have the Life values they received in PreRoll.. The first particle created in PreRoll was created at time 0.0.. By the time the anim actually begins, it has aged to a value of 1.0, so it expires and is replaced with a new particle.. In this example, since the PreRoll path and the regular path for the Fountain are the same, it is created at the exact position that the old one expired.. Let's verify this.

Set the Attach Point to some point on the circular path.. Use the Freehand Poly tool to draw a simple triangle of any convenient size.. Make this triangle a path and turn on Movement Status and Last Segment.. Assign the Fountain to this new path.. Now what you have is the Fountain PreRoll path

setting up the pattern you have seen, and the triangle moving the Fountain during the anim. Preview this. As the particles set up in PreRoll, age, and die, new ones are created along the new path.

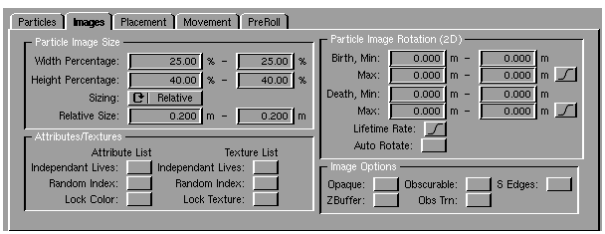
Edit the Fountain and change the Life values to 2.0/2.0, 2.0/2.0 and preview this. The PreRoll path has moved the Fountain during PreRoll and set up the pattern. During the regular anim, the triangle path is moving the Fountain stem, but since it has already generated its 200 particles during PreRoll and they live long enough to never die during the anim, they stay in their PreRoll positions.

/// Move Type doesn't exist 0430 /j
@ Edit the Fountain and select the Follow Path checkbox under the Placement Tab.turn the ///Move Type from ///Independent to
@ ///Grouped and preview.
@ if you want a Fountain to set up a pattern and then move the
@ pattern as a whole during an animation. One important aside, however:
@ If you want to see the pattern being built and then
@ move the pattern as you just saw, you would have to either do it
@ as two separate animations and then join them, or use two
@ Fountains, one to draw the pattern, the other to move it.

IMPORTANT!! PreRoll has no abort. When a Fountain is PreRolling, you must wait until it is finished. The waiting time varies based on the number of particles, and the "shortness" of their lifetimes. The more particles you ask for and the shorter their lifetimes, the longer PreRoll takes.

Now it's time to look at the render-related options, Image, Size, Color, Image Rotation, and Obscuring.

Particle image



Choosing a Particle Image is simply a matter of choosing a Texture List. The Texture List member(s) MUST use a bitmap. Procedural textures are not permitted for Fountains (or Flares) and will be ignored.

Normally, use the Flare images provided with the program, or custom images you have created just for Flares and Fountains. You should use a Color type for the members. Others may give the same results, but are not necessary. The Fountains only need a Color component from the members.

You're free to use a previous member with a strength modification (Alpha Channel). The Texture List can have as many Members as you wish, and will composite nicely. Keep in mind, however, that since Fountains are built additively, you may want to lower the Strengths of the members as you add more members.

The color slider in the Edit Texture List requester is treated as the amount of color in the Flare image to replace with the current member. If it is set to 1.0, the particles will have the Color of the current member. If it's at 0.0, the Color contributed by the current member will not be used. Its Strength will still be used, but the Color of the previous member will not be altered. If the previous member had no Strength at an area where this member does have Strength, the base color of the Fountain, as read from its Attribute List will be used.

The Particle Image bitmap contributes the Strength and Color information to the particle. The color is derived from the bitmap directly (and can be reduced with the color slider in Texture Member Control

requester).. The Strength is based on the maximum "Luma" at each pixel of the bitmap - the Luma is the maximum value of the red, green and blue color components.

Particle size

There are three methods of altering the size of the rendered particles.. The first is a carry over from Flare imaging, the second is by using the bitmap indices in the Texture List Members, and the third has been developed specifically for Fountains.. We will discuss the first and third, since you have already learned about bitmap indices.

The Flare carry over method:

/// This doesn't look like it's the same as in v<5.0 ??? 0421 /j

In Fountains, as with Flares, the controlling gadgets Flare Width Percent, Flare Height Percent, the Relative Distance establish the Size of the Fountain.. In the Particle tab, the size of the particles themselves is established in the Particle Relative Size panel.. The program images the particles as though they were polyspolygons of this size in space.. This means you get true parallax and perspective.. This is the normal way to use set up a Fountain.. You then use Width/Height to control the aspect ratio of the particle image.

The Fountain specific method

The Particle Relative Size gadgets in the Particles tab are the familiar Orig Max/Min and Dest Max/Min, along with their CSpline controls.
give you complete control of the particle image size.. Normal use is to change the multipliers to alter the size of the Fountain's particles.

The larger the particles, the longer they take to render.. It is quite possible to have particles that are as large as the screen or larger.. If there are a few hundred of these, it is just like asking the program to average and composite a few hundred full size pictures - and can take quite a long time.. Also, as the camera moves close to a particle, it becomes larger on screen, even with a normal particle size.. So, it's possible to have frames where the camera comes in close to particles that will take significantly longer to render than the ones where the camera is at more normal distances.

Particle color

In the Fountain's simplest use, particle color comes from the particle image you are using.. However, Aladdin 4D has extensive controls on altering and enhancing this color.

First, if a member of the Fountain's Texture List has its color at full value (1.0), the color of the particle at the areas of some strength of this member will be the color of the the member's bitmap. (This can be altered or replaced by succeeding members in the Texture List if they have some strength in the same area.)

Second, if a member of the Fountain's Texture List has its color slider at less than full value (less than 1.0), the color of the particle at the areas of some strength of this member will be the color of the Flare previous to this member, plus the color value setting of the member.. So if the Texture Member has some strength,

and the color is at no value (0.0), the particle will have the strength of the Texture Member, but the color will be that of the Attribute List, or the color modified by a previous Texture Member.

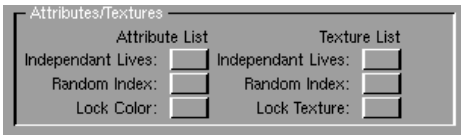
If you have the Texture List member(s) color set so that some of the color can be contributed from the Attribute List, you can choose among these possibilities:

First, the Fountain will index the Attribute List based on the animation's global time. The color of all particles will be that of the Attribute List at the current frame of the animation.

Second, you can turn ON Attribute List / Independent Lives in the Images tab in the Fountain Control requester. If this is ON, the Fountain will send each particle through the Attribute List completely during the particle's lifetime.

If you have Attribute List/ Independent Lives ON, you can choose to lock (see the Lock Color gadget) the particle into the color it found at the time it was born. If a particle is born at 0.1, it will look at the Fountains Attribute List and set its color to what the Attribute List has at 0.1. This color will then stay the same throughout the particle's Life.

With Attribute List Independent Lives is ON, you can instead turn Random Index on, to randomize the particles' color. This is important if all particles are born at the same time and you want some color differences among them.



A corresponding set of gadgets under Texture List, in the Attributes/Textures panel of the Images tab, applies Independent Lives with Random or Locked texture.

Particle strength

There are three distinct methods of controlling the strength of particles emitted by a Fountain. The first is in the Texture List member's Strength gadget. The second is an overall ##Fountain strength,

~~## don't see this anywhere in the fountain requester 0421 /j~~

and the third is Orig and Dest values for the individual particles.

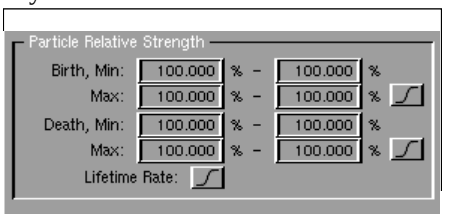
First: You can alter particle strength by lessening the strength of the Texture List member. Normally this strength is 1.0, and the ##Fountain-specific Strength gadgets are what you'd use to change the particles' strengths.

Second: The ##Overall Strength gadgets on the right-hand side of the Particles Tab~~at the bottom of the~~

~~## I only see Particles Relative Strength 0420 /j~~

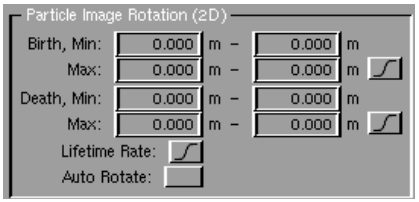
~~left column~~ let you can enter Entry/Exit strengths for the Fountain and control the rate of change between them with a CSpline. All particles in the Fountain will be affected by these values.

Third: The Particle Tab's gadgets let you edit Particle Relative Strength values for individual particles over time in an animation, and over the life of an individual particle.



It's possible to enter values greater than 100%. If you set one of the Strength values to greater than 100%, the final value that is to be composited will still be clamped to 100%. You can, however, promote an image to fewer levels of transparency in this manner. However, for normal use, set the Strength values between 0% and 100% and use CSplines to alter rates of change.

Particle image rotation

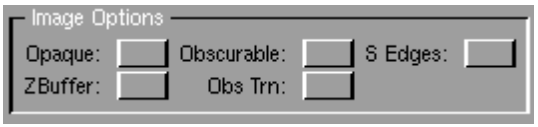


In the Images tab, the Particle Image Rotation gadgets control what you might infer from their name - image rotation in a 2D manner from birth to death.

The Max/Min values are limits to a random value assignment. The values are in degrees around the apparent axis of the particle image.

In the Particle Image Rotation (2D) panel, Auto Rotate orients the particles. If this is ON, the particles will automatically rotate so their "bottom" always points toward the source of the Fountain. This rotation is additive to the values you enter in the rotation gadgets, so you can use both.

Particle obscuring



Five options control how particle images interact with polygons and the screen in a drawing. These are the four Boolean gadgets labeled Opaque, ZBuffered, Obscurable, S Edges, and Obscured Trn.

Opaque: If this is ON, the fountain will behave as a transparent object. See Transparent/Opaque Fountains below.

ZBuffered: If this is ON, the program will intermix the particle images with the polygons in the drawing. This allows polygons to cover up only a part of a particle image. If the ZBuffered option is ON, the particles are not painted as a separate step.

Instead, the drawing will begin immediately (after PreRoll, if it's on) and the calculations and drawing of the Fountain are done on the fly as the scanlines that contain particle images are rendered.

If this option is ON, the program will have to create a buffer for every screen scanline that has particle images on it. In fact, a buffer for every particle image on a particular pixel on a scanline has to be made and maintained. This can take a LONG time for all the allocation and initialization, especially if the particles are all near each other, as at the beginning frame of a standard Fountain set up. In general, if you're going to use the ZBuffered option, you probably DO NOT want to use a large number of particles.

Obscurable: If this is ON, the program will test each particle against all the polygons in the drawing. If a particle is behind a polygon, it will not be imaged. This is akin to the way real world lens flares behave, and is suitable for almost all Fountain drawings. It is the suggested method of obscuring Fountains.

S Edges: If this is ON, the program will test each particle for being outside the screen boundaries. If it is outside, the image will not be rendered. If off, the image will render, even if the particle is off screen.

Obscured Trn: If this is ON, the program will test each partile against all the polygons in the drawing. If a particle is behind a polygon, it will then test the polygon for its transparency level (as set in the Attributes List, not from opacity maps). If the poly is transparent, it will lower the strength of the particle

image, reducing it to 0 strength as a square of the transparency. In other words, the particle will show through the poly according to the poly's transparency setting.

Watch out! This option requires separate transparency buffers for each image on each pixel on each scanline. This can cause a heavy memory demand when a lot of particles are in use. It can also cause extended rendering times, even with a moderate number of particles all in the same, or nearly the same place on the screen.

Transparent/Opaque Fountains

Flares and Fountains are frequently rendered as "additive" objects - their colors are added to the current color of the image at their location. You can have it either way, however. Aladdin 4D's additive mode is the default. But the "Opaque" option lets you render Fountains as transparent, instead. If you turn ON the Opaque gadget, the Fountain will be treated as a transparent object that you can use to create overlapping Flares and Fountains that partly or fully obscure other Flares, Fountains, and polyspolygons in the drawing. It also means you can create Fountains as dark objects if you want.

To have the Opaque option, you must use the ZBuffered option, too - if you select the Opaque gadget, the ZBuffered gadget will be automatically turned on for you. With Opaque ON, Fountains are treated as virtual polygons and added to the polygon base for the render, as with the Opaque option in Flares.

If you want a "cookie cut" for your Fountain when using the Opaque option, there are two ways to achieve this. First, you can set the Texture for the Fountain to be Decal or Genlock. This will use "Color 0" of the image for a "no image" zone, giving the cookie cut effect. However, this effect will give you aliasing.

To avoid this aliasing, you can put an Alpha Type in the Texture List and use it to modify the image of the Fountain. This results in Fountain images that are of great quality regardless of their size.

See the Flares section for the discussion of triangluar shaped confetti. (#Xref confetti) The process for a Fountain is the same.

Alpha Channel

Aladdin 4D implements an "Alpha Channel" in texturing, so you can modify one texture with another. The Alpha Channel texture type is like the alpha channels used in some display hardware. If a texture member is this type, it modifies the strength of the next member based on the luma value at a given pixel. This means you can cookie-cut the next Texture Member, and even use animated cookie cuts.

As with other "non-Color" types, the color slider is interpreted in the Alpha Texture Type to mean "add color". Set it to 0.0 if you do not want to see the texture, but just have it affect the next member. This type will not work with bump, reflection, or genlock maps. It is also not engaged for foreground textures, since foregrounds are either on or off. It does, however, work with backgrounds and overlays. Its application with overlays is particularly useful.

Reference Section

The information in this section is listed approximately in the order in which the items occur in the program. Most of the menu items and tools are indexed, so you can check the index for quick pointer to further discussion in the tutorials or general information sections.

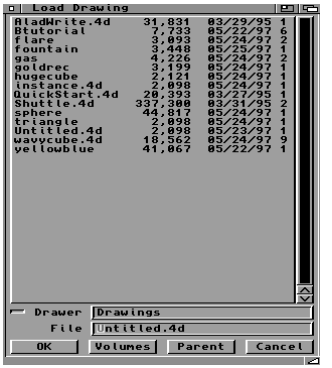
Pull-Down Menus

The Project Menu

Menu Item: New
Keyboard: None

Brief: This item deletes all spaces and polygons in use. It clears all attribute and texture lists, and you are then presented with a totally new space in which to work.

Limits: You cannot use this menu item during facing or deform editing.



Menu Item: Open
Keyboard: <F5>

Brief: Use this menu item to Open (that is, load) an existing file, translating as necessary, and selecting via the standard Amiga file requester.

You can load, re-load, and open an existing file for editing, rendering, etc. If you load a file, anything in use at the time is replaced by what you load (See Append). Aladdin 4D can import a variety of file types, with varying results as follows:

Aladdin 4D and Draw4D formats

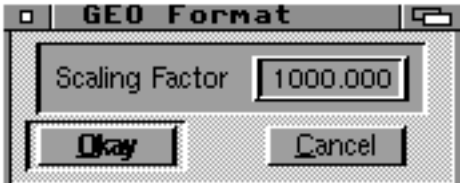
Aladdin 4D files: Clears current project completely. Loads file including all of its spaces, polygons, shading, textures, cameras, backgrounds, attribute lists, etc. If you load files which were saved by versions of the program before 2.0, any eases and delays are automatically translated into CSplines if they were in use.

Draw4D-Pro files: Clears the current project completely. Loads file including all spaces and polygons. Shading and texture assignments are not compatible and are ignored. You are asked, however, if you want any bitmaps that were used as textures loaded for re-application in Aladdin 4D, including the background. Paths are created and linked as used in Draw4D-Pro, but with only a single level of rotation since that's all Draw4D-Pro supported. Lights are loaded, but are given default attributes. A new Attribute List should be created and assigned. EyePaths are ignored, although the path is loaded and a camera may be assigned. Eases and delays are automatically translated into CSplines if they were in use.

Draw4D files: This format is also recognized. Polygons are loaded along with any paths and path assignments. No color information is loaded. No shading or texture information exists in these files, so

must be created after loading.. Eases and delays are automatically translated into CSplines if they were in use.

VideoScape .Geo files



The .geo (VideoScape) file format is recognized and can be loaded.. Polygons are given default Attribute List.. No shading, color or textures are loaded and must be assigned in Aladdin 4D.. No camera or lights are read.. Groups are established according to like colors, to facilitate handling of polygons.. You are asked for a scaling factor whose value depends on the originating program.. For most files, a factor of 1000 is a good starting place.. If the object loads too small, reload it with a larger factor.. If it's too large, either reload it with a smaller factor, or resize it in Aladdin 4D.. On loading, you're asked if you want to eliminate any doubles (duplicate polygons are required in .geo files, but not in Aladdin 4D).. It is good practice to do so.. You can also choose to eliminate any single points and straight lines (detail ~~polys~~polygons).

Scenery Animator DEM files

These Digital Elevation Maps are recognized and loaded.. You are presented with a requester allowing you to specify the sampling frequency of the digital grid in the file.. You are also given the option to simplify the resulting triangle information to rectangles and a variable. "Simplify angle" considers the cosine of the angle between two adjacent triangles.. When it is less than that specified, they will be joined into a single rectangle.. You may also choose a platform.. This is a set of polygons around the rim of the land segment that descends to sea level.. Based on your choices, the requester lets you know how many polygons will be created.

Encapsulated PostScript (EPS) files

The Encapsulated PostScript file format is recognized from a few specific drawing programs.. On the Amiga, these include Professional Draw up to version 3.0, and ProVector files up to version 2.1.

EPS files produced by other versions of these, or by other programs may also be recognized, so just test them to see.. Although the EPS format is intended to be cross-platform portable, in reality of course it is not.. To find out if an EPS file works, just try it!.! Some non-Amiga EPS files will be recognized, although some EPS files that originate from Amiga programs might not.

When you open an EPS file with Aladdin 4D, you will be presented with a "Load EPS" requester.. Here you can decide how you want the file translated.. The EPS file contains information for Bezier curves.. These must be turned into multi-sided curves for Aladdin 4D.. If you turn "Relative" ON, the number of sides for a Bezier curve will be adjusted - smaller for short Beziers, larger for long ones.. If you set Relative OFF, all Beziers will get the same number of sides regardless of their length.. Resolution determines the number of sides to be used in a default length Bezier and is a gauge of resolution, not an absolute.. You may also choose to remove any excess points that are collinear by turning "Remove Pnts" ON.. The angle is presented in degrees, and any points that form an angle less than this will be removed.

You can help out the loading process if you have control over the program that generates the EPS file.. Before you print your drawing as an EPS file, convert any ellipses to Beziers - ellipses will not be translated.. Watch out for other special forms, such as grids, compound objects, etc., that also may not be translated properly.. Some non-Amiga EPS files embed text objects as text, but may or may not supply the font vectors for the text.. If you can, turn all text into outlines before generating the EPS file, so any text objects will have a better chance of moving over to Aladdin 4D, without having to port any odd fonts.

LightWave Objects

LightWave objects are also loaded. They are added to the scene in Aladdin 4D.

Menu Item: Append
Keyboard: <Shift> <F5>

Brief: This menu item will open an existing Aladdin 4D file and create additional spaces in the current drawing for it. The menu brings up the Load Object requester, in which you select the object to append. The program translates as necessary. If you have three spaces in your current drawing and load a drawing with three spaces in it, you will have a drawing with six spaces. You are then free to jump the objects that you want from the file into the space you want. The result is that you have both drawings as one, to edit as you wish.

When you use Append, all Texture Lists, attributes, shading, etc. are loaded and assigned. Note that if one of the Texture Lists in the new file uses a bitmap that you have already loaded, it will use it, not duplicate it. If a background, foreground or overlay is found, you will be asked if you want it.

Menu Item: Save
Keyboard: <Shift> <F6>

Brief: This menu item saves your current project under the current name, replacing the old file of the same name. If you have not yet named the project, the file requester will open asking for a name.

Menu Item: Save As... (sub-menu)
Aladdin 4D (requester: Save Aladdin 4D)
GEO (requester: Save .GEO)
Keyboard: None

Brief: This item lets you save a new project, or save an old one by a new name. There are two sub-items to this menu, defining the file type you wish to save:

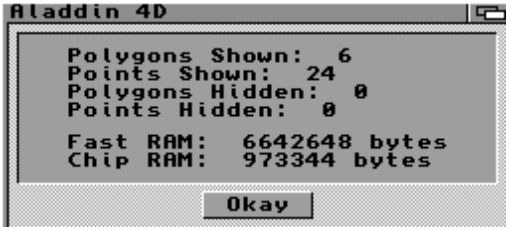
Save As / Aladdin 4D: Opens the Save Aladdin 4D requester asking you for a path and name to save the file. The file is saved as an Aladdin 4D file.

Save As / GEO: Opens the Save .GEO requester asking you for a path and name to save the file. You are asked for a scaling factor; 1000 is recommended unless you have a specific target program in mind and know the proper scaling factor. The file is saved as a .geo (VideoScape) file.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Tools (sub-menu)
DPaint
ImageFX
Keyboard: (None)

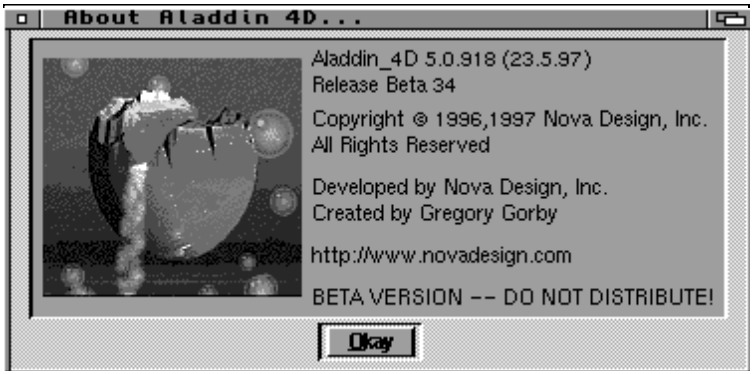
Brief: This menu item is here to give you quick access to paint and image processing programs of your choice. The sub-items are dynamically loaded from scripts that you can edit and rename to your liking. By default, the item fires up the paint program DeluxePaint (Electronic Arts) and the image processing - and paint - program ImageFX, if you have them.



Menu Item: Information
Keyboard: None

Brief: Shows total points and polygons for current space.

When selected, you will see an informational "Aladdin 4D 5.0" window about the current project, including the number of polygons and points in the current space. Also displayed is the amount of memory you have available. Some special polygon types are not represented.



Menu Item: About
Keyboard: None

Brief: This menu item displays version and other information about Aladdin 4D. Copyright information is also shown.

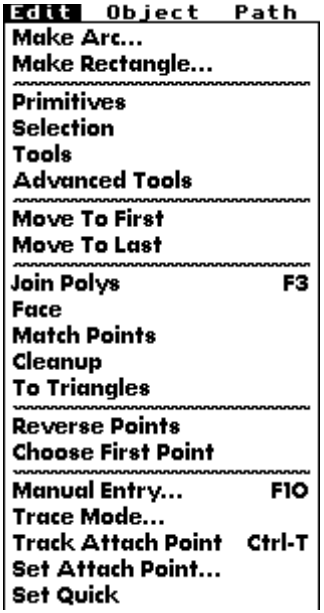
The illustration here shows the about requester from a beta version of Aladdin 4D.

Menu Item: Quit
Keyboard: None

Brief: This menu item will quit the program. You are asked to verify that you wish to do so before continuing. Always use the Quit menu item to close Aladdin 4D, so the program can record the settings for your tools, window preferences, etc. The program restores these settings on startup, even after power off.

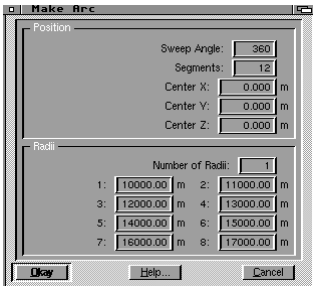
Limits: You cannot use this menu item during facing or deform editing.

The Edit menu



Menu Item: Make Arc
Keyboard: None

Brief: Create an arc shaped polygon.

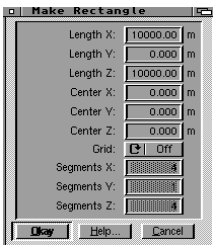


This menu item allows you to create any portion of an arc in any resolution. When selected you are presented with the Make Arc requester, where you can choose how many degrees of arc you want - the sweep parameter. A 360-degree arc is a full circle, of course, so if you want half of one, enter 180. For a third of a circle, enter 120, etc. "Segments" specifies the resolution of the arc - how many straight-line vectors are to be used to draw it, in other words. Center X, Y, and Z specifiers determine the position of its center in space - though of course you can just move it where you want after it's created.

You can also choose an arc of multiple radii - that is, an "s" shape. The Radii section of the requester lets you specify first, the number of Radii you want, and then the specification for each actual radius, up to 8 Radii, total.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Make Rectangle
Keyboard: None



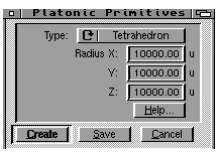
Brief: Create a rectangular polygon.

This menu item allows you to create a new polygon that is either a rectangle or a rectilinear solid. When selected the Make Rectangle requester opens, allowing you to specify the size, position and a grid option. The Length X, Y, Z specifiers do the size; Center X, Y, and Z do the position, and if you turn GRID on, the Segments boxes become enabled, to let you specify a grid to be applied to the resulting object.

As with most of the menu items in Aladdin 4D, when you set up the Make Rectangle requester, its settings can be saved (use the Save button) as defaults for future use of the option.

The rectangular forms illustrated are typical. This tool normally groups in a staggered fashion so you can easily create checkerboard color arrangements.

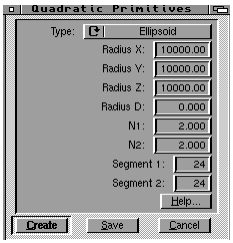
Menu Item: Primitives (sub-menu)
Platonic Primitives
Quadratic Primitives
Keyboard: None



Brief: Create a basic - "primitive" object for editing.

This menu item performs the same function as the prim-a and prim-q tools in the tool box. It creates a basic mathematical shape that you can use singly or in combination to model the shapes you want. Select the sub-menu for the type of primitive shape you want, simple geometric (Platonic Primitives) or more complex (Quadratic Primitives).

Platonic Primitives are: Tetrahedron, Icosahedron, Octahedron, and Dodecahedron. The requester has Radius X, Y, and Z gadgets to let you specify the radius for your object.



Quadratic Primitives are:
Hyperboloid
1, Hyperboloid 2, Torus, Elliptic Cone, Elliptic Hyperholoid, and Ellipsoid.

In the Quadratic Primitives requester that opens, Radius X, Y and Z are the size. Radius D is the size specifier for primitives which have a "hole" in the center, like a torus, expressed as a percentage of the main radii. This must be positive, and a normal range would be 0.0 (no hole) to 5.0.

N1 and N2 control the "roundness" of two aspects of the quadratic. A value of 2.0 gives a circular arrangement, 1.0 gives a linear appearance, and 0.5 gives a "coved" arrangement. It is best not to use values below 0.5, or duplicate ~~polys~~polygons can be created as well as ~~polys~~polygons with multiple collinear points - which can cause rendering artifacting. Values higher than 2.0 give a "rounded corner" appearance and are quite useful. Segment 1 and Segment 2 control the number of polygons generated. Segment 1 controls divisions around the rotational axis (like in a Lathe), and Segment 2 controls the divisions around the opposing axis.

You may want to convert the generated primitive to triangles before rendering, especially if the ~~polys~~polygons are not planar.

The generated primitives are grouped in a natural way for your convenience in shading. In Group 1, they are grouped as a whole. In Group 2, they are grouped in quadrants, other levels have different arrangements.

Limits: Be careful not to set N1/N2 to values that are too low for the size object you are creating.

As with other Aladdin 4D requesters, the Primitives ones permit saving the parameters you set up, for later re-use of the menu sub-items (or tools). Create makes the object you've defined. If you don't click save, the defaults will come back without your modifications next time.

See illustrations of the objects in the Tools reference sections at Prim-Q.

Menu Item: Selection (sub-menu)
Multipoint...
Select Same...
Keyboard: None

Brief: This menu item performs the same function as the Multi Point tool in the tool box. It lets you select multiple objects by your choice of methods. Generally, you'll use this menu item to set up the tool for use, though you can also click the right mouse button over the tool's icon to bring up its Settings requester.

Multipoint selection lets you work on an arbitrary group of points as though they were a single object or polygon. The ~~rotary gadget~~cycle gadget lets you choose Single, Box, Lasso, or Polygon modes,

/// TEK: Nice, very nice. I'm so glad we paid $1000 for this kind of professionalism.
Clear All Points..~~.~~ ~~well, it~~ it clears all points.

The Select Same sub-menu lets you select ~~polys~~polygons based on what they are, contain, or have assigned to them~~.~~. For example, you can select polygons that share the same Texture List, the same Attribute List, etc~~.~~. tool default window~~.~~. The gadgets in this window are:

Same Attributes**:** Will select all ~~polys~~polygons in the current space that use the same Attribute List as the currently selected poly~~.~~. Requires a single selected poly.

Same Texture: Will select all ~~polys~~polygons in the current space that use the same Texture List as the currently selected poly~~.~~. Requires a single selected poly.

Same Path: Will select all ~~polys~~polygons in the current space that are assigned to the same path as the currently selected poly~~.~~. Requires a single selected poly.

2-Point Polygons: Selects all ~~polys~~polygons in the current space that have only two points.

Complex Polygons: Selects all ~~polys~~polygons in the current space that have more than four points.

Duplicate Points: Selects all ~~polys~~polygons in the current space that have duplicate points.

Triangles: Selects all triangular ~~polys~~polygons.

Paths: Selects all ~~polys~~polygons in the current space.

Special Objects: Selects all special objects - like Flares and Fountains.

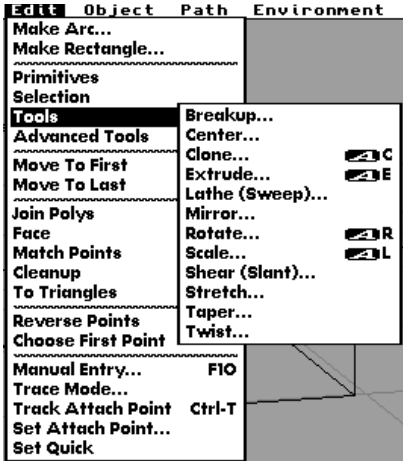
Lights: Selects all lights in the current space.

Deselect First: will deselect any ~~polys~~polygons that are already selected before selecting ~~polys~~polygons satisfying the request.

Hide All But Selected: will automatically hide any ~~polys~~polygons not chosen when satisfying the request.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool)~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the defaults will come back without your modifications next time.

Edit /Tools Menu



Edit / Tools Menu Item: Break-up
Keyboard: None

Brief: Simplify complex polygons~~.~~ This menu item performs the same function as the Breakup tool in the tool box~~.~~ It allows you to break a complex polygon down into smaller ones.

The requester it opens is: Breakup Settings, containing only the option to delete (or retain if unchecked) originals.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool)~~.~~ Create makes the object you've defined~~.~~ If you don't click save, the defaults will come back without your modifications next time.

Edit / Tools Menu Item: Center
Keyboard: None

Brief: Center the view, or selected ~~polys~~polygons.

This menu lets you establish settings for the Center gadget, or perform the operation, as you choose.

The Center Settings requester offers:

Center X, Y, and Z - normally the origin, but you can make it anywhere you like.

The Current ATP button establishes the "center" as the current Attach Point.

Percent X, Y, and Z let you specify where you want the center in percentages of the available space along each of the axes.

Center Around (check boxes) X, Y, Z: Specifies an axis to perform the center operation relative to.

Single Polygon: Tells the program to do the operation on only a single poly.

Polygon Groups: tells the program to center groups.

Center Deforms: tells Aladdin 4D to perform the centering operation on any Deforms that have been established.

Limits: None

Edit / Tools Menu Item: Clone
Keyboard: <Amiga> <A>

Brief: Create a duplicate of the currently selected object, path, etc.

This menu item performs the same function as the Clone tool in the tool box~~.~~. It copies - clones - an object, polygon, path, whatever.

The Clone Settings requester which opens lets you specify Offset X, Y, and Z for the copy - so the new object won't be on top of the one you're copying.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool)~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the defaults will come back without your modifications next time.

Edit / Tools Menu Item: Extrude
Keyboard: <Amiga> <E>

Brief: Project the currently selected poly into another dimension, creating additional ~~polys~~polygons and groups as appropriate.

This menu item performs the same function as the Extrude tool in the tool box~~.~~. It creates an additional dimension onto an object by extruding it in whatever direction you choose (and by whatever amount you specify).

The Extrude Settings requester which opens lets you specify Length X, Y, and Z for the extrusion operation~~.~~. You can get this requester by clicking the Right mouse button over the Extrude toolbox gadget, or just pull down the menu.

You can also change Final Size in each of X, Y, and Z Axes~~.~~. Final Size controls any desired reduction or enlargement as the extrude occurs.

Segments is the number of segments the extruded object is to have~~.~~. It should be non-zero.

Swell: Causes the extrusion to occur according to a power function, enabling easy creation of some useful shapes~~.~~. A Power must be set in the Swell Power gadget for this to have any effect, and Far and Near options (in the ~~rotary gadget~~cycle gadget) permit directing the Swell operation.

Swell Power: The strength of the Swell operation (see above).

Connect Last Segment: The Connect Last Segment switch tells the extruder whether the segment of the selected polygon between the first and last points should be extruded~~.~~. For instance, if you extrude a square with this on, you will get a square pipe~~.~~. If it is off, you will get a pipe with one side missing, like a rain spout.

Delete Concurrent: If ON, after the extrude is performed, all new ~~polys~~polygons will be examined and if any are in exactly the same space, they will be deemed to be interior to the extrude and will be removed~~.~~. This is very important for fonts, which often have lines bridging the outside of the letter with the inside, like the letter "O"~~.~~. If you extrude this letter with Delete Concurrent OFF, you will get two extra ~~polys~~polygons at this "seam", and they will play havoc with the shading routines~~.~~. In general use, leave this gadget ON.

Poly To Poly: If this is ON, the program will look through all the ~~polys~~polygons in the drawing and determine whether you have exactly two ~~polys~~polygons selected~~.~~. If this is true, it will count the points of both ~~polys~~polygons to determine whether they are the same~~.~~. If this also is true, it will create a shell between the two ~~polys~~polygons, on a point for point basis, starting with their first points~~.~~. This allows you to create "cross sections" of objects, and then "skin" them with a single action.

Single Group: This option makes the extruded a single, rather than multiple group.

If you don't like the extrude produced use the "DeletePoly" tool before setting the changes, to retain the original shape.

If the extruded ~~polys~~polygons are assigned to paths, the new ~~polys~~polygons will automatically be assigned to the same path, unless the path uses deforms.

Some typical examples of extrudes can be found in the external tools section later in this manual.~~:~~

~~Offsets: 0 10000 0~~
~~Final Size: 100 100 100~~
~~Segments: 1~~
~~Connect Last Segment: ON~~
~~Swell: Near~~
~~Swell Power: 0.0~~
~~(Attach Point at top point)~~

~~Offsets: 0 10000 0~~
~~Final Size: 10 10 10~~
~~Segments: 8~~
~~Connect Last Segment: ON~~
~~Swell: Near~~
~~Swell Power: 2.0~~
~~(Attach Point at center of poly)~~

~~Offsets: 0 10000 0~~
~~Final Size: 10 10 10~~
~~Segments: 8~~
~~Connect Last Segment: ON~~
~~Swell: Far~~
~~Swell Power: 2.0~~
~~(Attach Point at center of poly)~~
~~Offsets: 0 10000 0~~
~~Final Size: 10 10 10~~
~~Segments: 8~~
~~Connect Last Segment: OFF~~
~~Swell: Near~~
~~Swell Power: 2.0~~
~~(Attach Point at top point)~~

~~Offsets: 0 10000 0~~
~~Final Size: 50 50 50~~
~~Segments: 8~~
~~Connect Last Segment: ON~~
~~Swell: Near~~
~~Swell Power: 1.5~~
~~(Attach Point at center of poly)~~

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool). Create makes the object you've defined. If you don't click save, the previous defaults will come back without your modifications next time.

Edit / Tools Menu Item: Lathe (Sweep)
Keyboard: None

Brief: Create a three-dimensional object by sweeping a two-dimensional one through one of the other axes.

This menu item performs the same function as the Lathe tool in the tool box. A most useful tool in 3D modeling, it's a rotational extrude or "Lathe." This allows you to create all types of "round" forms from a single polygon by rotating the polygon around a selected point in space and creating an extruded shell along the points of the polygon as it rotates. You need only draw the cross-section to obtain a rounded, lathed object.

When you select this menu item, you open the Lathe settings window. The gadgets are:

Sweep Around: A ~~rotary gadget~~cycle gadget with which to pick X, Y, or Z axis.

Sweep Angle: The size of the arc the Lathe will use in degrees. It can be negative if needed. If you use offsets, it can be greater than 360.

Sweep Points: This is the number of divisions the Lathe will produce. Use a larger number to obtain smoother shapes.

Connect Last Segment (check box off or on): This determines whether the last side of the poly being swept will create ~~polys~~polygons in the shell. Connect Last Segment tells the program whether to Lathe the last edge of the poly (between the first and last points). Doing so will create a central "stem" in the object. If you make an arc of 180 degrees and spin it around one of its end points, you get a sphere. But if you do the Lathe operation with the last segment ON, you get a "stem" running through the center of the sphere. This stem will cause shading oddities. For a closed form, like a sphere, you should turn Connect Last OFF. If, however, you are Lathing a shape into an open form, like a donut, and have the Last Segment OFF, you will produce a donut with an opening running all around it. For some purposes - making an automobile tire, for example - you might want the open shape, even though it's not to be part of a torus. For an open form, like a torus, you generally want Last Segment ON.

Solid: The Solid switch allows you to do a "rotational clone". It is very useful for duplicating a shape around an axis, like the fins of a rocket. However, this is not really a clone tool and will not duplicate textures or shading. These must be applied to the forms after their creation.

Final Size X, Y, and Z: These parameters allow you to do an enlargement or reduction of the original shape during Lathe operations.

Final Position (~~Rotary Gadget~~Cycle gadget): This gadget lets you specify where the lathe will stop. The choices are: Outside, Start, Center, Inside

Offset X, Y, and YZ: The Offset boxes allow you to specify a translation of the shape as the Lathe operation occurs.

Group Skip: If set to other than 1, the created ~~polys~~polygons are grouped in alternating manner based on this number.

The shape that you use the Lathe on is NOT deleted, and generally should be selected and deleted, or moved out of the form when the lathe operation is finished.

Limits: Lathe operations may occur only around main axes.

Some typical examples of Lathe operations and the shapes that made them are shown in the discussion of this feature in the external tools section.~~:~~

~~Sweep Angle: 360~~
~~Sweep Points: 12~~
~~Connect Last Segment: OFF~~
~~Solid: ON~~
~~Final Size: 100 100 100~~
~~Offset: 0 0 0~~

~~Sweep Angle: 360~~
~~Sweep Points: 12~~
~~Connect Last Segment: ON~~
~~Solid: ON~~
~~Final Size: 100 100 100~~
~~Offset: 0 0 0~~

~~Sweep Angle: 1080~~
~~Sweep Points: 48~~
~~Connect Last Segment: ON~~
~~Solid: ON~~
~~Final Size: 100 100 100~~
~~Offset: 0 0 20000~~

~~Sweep Angle: 720~~
~~Sweep Points: 24~~
~~Connect Last Segment: ON~~
~~Solid: OFF~~
~~Final Size: 100 100 100~~
~~Offset: 0 0 24000~~
~~Sweep Angle: 720~~
~~Sweep Points: 24~~
~~Connect Last Segment: ON~~
~~Solid: ON~~
~~Final Size: 0 0 0~~
~~Offset: 0 0 0~~

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool)~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the defaults will come back without your modifications next time.

Edit / Tools Menu Item: Mirror
Keyboard: None

Brief: Create a mirrored copy of an object, or flip it.

This menu item has a two-fold purpose. First, it can flop an object left/right, top/bottom. Second, in the course of doing that, it can create a duplicate, new object, leaving the old one in place.

Mirror X, Y, or Z Axis (check boxes): This is the axis for the mirror

Mirror Around (Rotary GadgetCycle gadget): Choose a position for the mirror, either Center, or ATP (that is, the Current Attach Point).

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item. Create makes the object you've defined. If you don't click save, the previous defaults will come back without your modifications next time.

Edit / Tools Menu Item: Rotate
Keyboard: <Amiga> <R>

Brief: Rotate selected polyspolygons.

This menu item allows you to rotate all selected polygons. It opens the Rotate Settings requester. In the requester:

Angle X, Y, and Z tell Aladdin 4D how much to rotate and in what direction. Angles are in degrees. PolysPolygons must already be selected and you must use the Perform button for this to take effect when you exit the requester.

Rotate Around: The Rotate Around gadget can be either Center or ATP (Attach Point) This tells the program where you want the rotation based. If Center, the program rotates all selected polyspolygons around their common geometric center. If Attach Point the program will rotate all selected polyspolygons around the current Attach Point.

Rotate Deforms: Here you can choose whether you want to rotate any deforms that may exist for the selected polyspolygons at the same time you rotate the polyspolygons. THIS IS NOT UNDOABLE.

Rotation occurs in XYZ order.

Limits: None

Edit / Tools Menu Item: Scale
Keyboard: <Amiga> <L>

Brief: Change the size of selected polygons, objects, etc.

This menu item allows you to resize any selected polygons along any axis or combination of axes by any amount. The option opens the Scale Settings requester.

Scale X, Y, and Z: These specify a percent of former size. 100.00 % is no change. Be careful - if you resize very small, so that adjacent points on an object become the same, you cannot scale them back up again to reclaim the original object. This is, however, common practice for one level of a deform when you want an object to vanish in size during an animation.

Radial: This an Off/On toggle. If ON, the program will measure all selected polygons and describe a virtual sphere that will just enclose them. During the resize, the points of the polygons will be driven

toward (or away from) the nearest point of the sphere~~.~~. This mode can take advantage of "extender" ~~polys~~polygons~~.~~. Also see the extenders and how they are used in the Stretch tool illustrations.

~~@~~ Radial Distance: Distance beyond which points will not move; points beyond it will resize negatively, points before the distance will resize normally. ~~/// This is not mentioned in old docs 0422 /j~~

Scale From: This ~~rotary gadget~~cycle gadget toggles between Center and ATP~~.~~. If Center, the program scales all selected ~~polys~~polygons around their common geometric center~~.~~. If Attach Point the program will scale all selected ~~polys~~polygons around the current Attach Point.

Scale Deforms: Another ~~rotary gadget~~cycle gadget toggle (It reads No or Yes)~~.~~. If this is ON, any ~~polys~~polygons that are selected and have deform levels will have their deforms scaled at the same time as the ~~polys~~polygons~~.~~. If OFF, the deforms will remain unaffected~~.~~. This is NOT undoable.

Limits: None

See also the illustrations accompanying the Scale Tool in the Tools reference section.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the previous defaults will come back without your modifications next time.

Edit / Tools Menu Item: Shear (Slant)
Keyboard: None

Brief: Shear or Slant selected polygons.

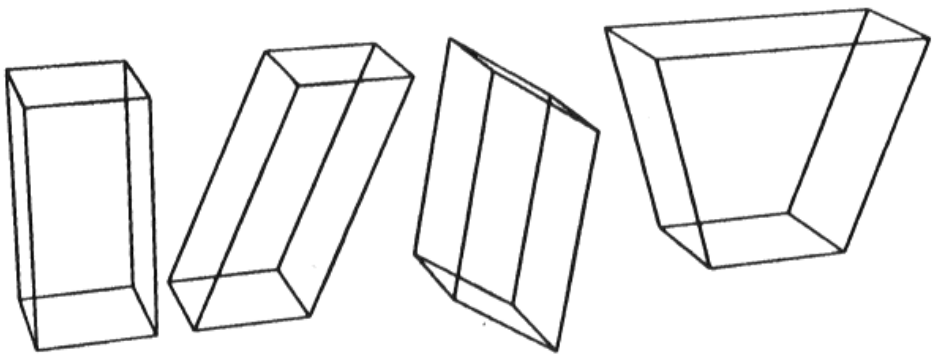
The Shear / Slant menu item allows you to skew selected polygons~~.~~. This can also be done interactively with the Slant tool~~.~~. This is useful for making pseudo italics when using fonts, or for putting the right skew on the wing of an aircraft or other objects~~.~~. The item brings up the Slant Settings requester.

Slant Type: This can be Linear or Mirror~~.~~. When using Mirror, be sure to set the Attach Point to the center of the mirror effect you want, or use the center options.

Slant From: This ~~rotary gadget~~cycle gadget is a toggle between Center and ATP (current Attach Point)~~.~~. This controls whether the slant is from the center of the selected ~~polys~~polygons, or the from current Attach Point.

In using this option, extender ~~polys~~polygons are sometimes helpful in placing the point you wish to move.

Limits: None



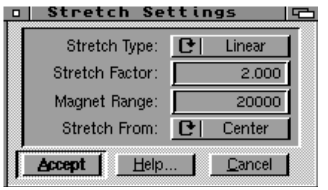
With the Shear/Slant tool, you can “pull” and object in any direction~~.~~. These forms were created using Shear/ Slant on a circle of 60 segments

It is sometimes helpful to use the Point Control tool to add points to a poly before using Slant, to obtain "rounder" results~~.~~. The Point Control tool can then again be used to eliminate collinear points in the final form.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the previous defaults will come back without your modifications next time.

The Shear/Slant tool works interactively~~.~~. You can set up the parameters you want with the pull-down menu, but you must operate the tool with the mouse to make it work~~.~~. Select the object you want to work on~~.~~. Be sure you have the view axis set up so the mouse’s motion will pull the object in the direction you want~~.~~. A little experimentation will work wonders for your understanding.

Edit / Tools Menu Item: Stretch
Keyboard: None



Brief: "Pull" selected polygons in a variety of ways in a given direction.

This menu item lets you s-t-r-e-t-c-h a polygon or a group of them this way or that, to achieve the shape you want~~.~~. The option opens the Stretch Settings requester.

Stretch Type: This ~~rotary gadget~~cycle gadget selects Linear, Mirror, or Magnet - the types of stretching you can apply.

If mirror, you should set the current Attach Point at the position you want the mirror to originate.

Magnet: Allows you to change the tool into a magnet tool. Note that when using the tool as a magnet, it is often desirable to draw a single line out from the main form. This is called an "extender poly" and is used to help lessen the effect of the magnetism by moving the point that is chosen to move some distance from the main form. As in nature, the effect of a magnet is greatly increased as the magnet gets closer to the form. Using an extender poly keeps this point of greatest affect some distance from the main form, allowing a smoother change.
~~@ Magnet: /// does what ??? 0422 /j~~

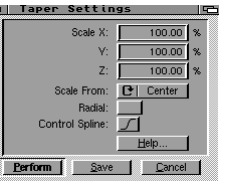
Stretch Factor: This is the amount of stretch to be applied~~.~~. The default is 2.000.

Magnet Range: How far the magnet "reaches."

Stretch From: This ~~rotary gadget~~cycle gadget toggles between Center and (current) Attach Point.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item~~.~~. Create makes the object you've defined~~.~~. If you don't click save, the previous defaults will come back without your modifications next time.

Edit / Tools Menu Item: Taper
Keyboard: None

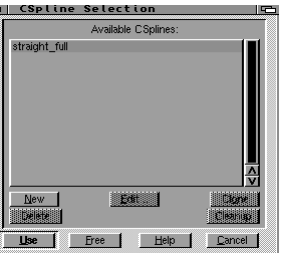


Brief: Alter the shape of selected polygons in a tapered manner.

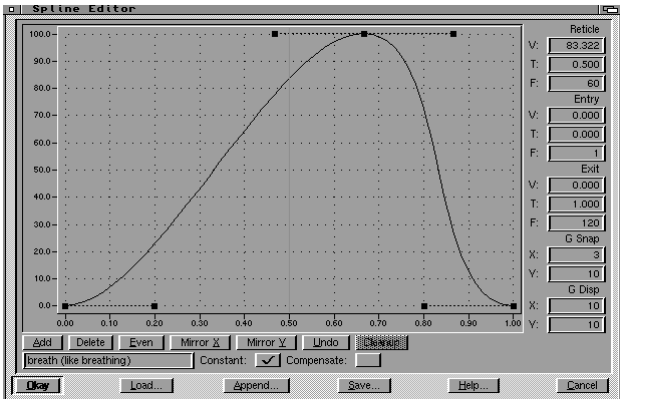
This option is similar to scale, but dedicated to tapering of the selected polygons or objects~~.~~. It opens the Taper Settings requester.

Scale X, Y, and Z: The percent of change along the respective axis~~.~~. Scale From: This ~~rotary gadget~~cycle gadget toggles between Center and ATP (current Attach Point)

Radial: Radial is either off or on~~.~~. If on, Aladdin 4D constructs a virtual sphere around all of the selected objects and applies the taper parameters to that.

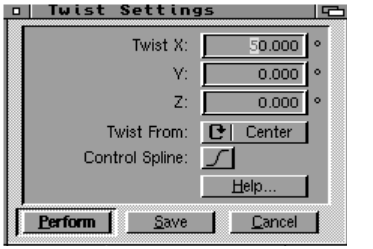


CSpline gadget: brings up Spline Selection requester listing available CSplines~~.~~. You can either select one that already exists or make a new one, clone an existing one, etc~~.~~. A CSpline can be used to taper objects in a very complex way that otherwise would require several scaling, stretching, shearing, and tapering operations.



As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item~~.~~. If you don't click save, the previous defaults will come back without your modifications next time.

Edit / Tools Menu Item: Twist
Keyboard: None



Brief: Create excellent twisted objects easily, like corkscrews and solid spirals.

Twist is another specialized application of scaling with a specific, complex function in mind~~.~~. It twists the selected ~~polys~~polygons, so you can easily make twisted objects without dealing with the complicated 3D math that would otherwise be required.

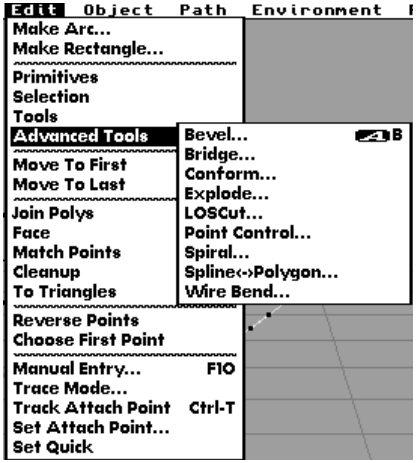
The Twist menu option brings up the Twist Settings requester.

Twist X, Y, and Z: These are the settings in degrees of the amount of twisting you want done.

Twist From: Toggles between Center and ATP~~.~~. If Center, the twist is performed relative to the geometric center of the selected ~~polys~~polygons~~.~~. ATP makes the twist occur at the Current Attach Point.

CSpline gadget: brings up Spline Selection requester listing available CSplines~~.~~ You can either select one that already exists or make a new one, clone an existing one, etc.

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item (or tool)~~.~~ Perform does the action you've defined~~.~~ If you don't click save, the defaults will come back without your modifications next time.

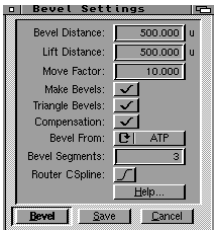


The Advanced Tools sub-menu

Edit / Advanced Tools Menu Item: Bevel
Keyboard: <Amiga> <B>

Brief: Make easy bevels on selected ~~polys~~polygons.

This menu option makes bevels easy~~.~~ The option opens the Bevel Settings requester.



Bevel Distance: How far the bevel should extend from the edge of the selected poly

Lift Distance: The amount of "rise" for the beveled surface.

Move Factor: This is only used when Bevel is in freehand mode~~.~~ The amount of movement of the bevel faces relative to the amount of movement of the mouse can be controlled by adjusting this value~~.~~ A lower number results in finer control~~.~~ The default value is 10.

Make Bevels: This toggle is either checked or not~~.~~ If ON, the bevel ~~polys~~polygons and bevel faces will be made~~.~~ If OFF, only the bevel faces will be made.

Triangle Bevels: If ON, the bevels that are created will be triangles~~.~~ If OFF, the bevels will be rectangles~~.~~ When beveling relatively small objects, the rectangles that are created may not be flat~~.~~ If when you render, you see some artifacting in the bevels, you can probably remove it by using the triangle tool on the bevels.

Compensation: Default is OFF~~.~~ If ON, the bevel distance will be altered by a factor based on the angle between two adjacent segments of the polygon~~.~~ This is a very subtle change, but an important one for some types of polygons~~.~~ If the shape of the bevel poly you create is not what you expected, turning this ON will often create the intuitive shape you had in mind.

Bevel From: This ~~rotary gadget~~cycle gadget toggles the bevel operation between Center and ATP~~.~~ This option is usually used as ATP, since that is the intuitive method of operation.

If ATP, the direction of the "lift" distance will always be away from the current Attach Point~~.~~ If Center, the direction of the "lift" distance will always be along the polygon normal.

This means that if the polygon normal for two letters in a font point in opposite directions, you must Bevel From ATP in order for the bevel lift to occur in the same direction for both letters.

To bevel a sphere, set the Attach Point to the center of the sphere first, then bevel it~~.~~ If you set the Attach Point to a point outside the sphere, the ~~polys~~polygons on half of the sphere will lift outward from the sphere, and the other half will lift inward~~.~~ You can bevel an object like a torus by setting the Attach

Point at the center, then beveling the "outside" half of the torus with a positive lift distance, then beveling the "inside half" of the torus with the same distance, but negative.

Bevel Segments: The number of divisions (or "poly rings") that will be created in the bevel. Be careful when setting Bevel Segments, as it can dramatically increase the number of polygons in the drawing.

This value is ignored unless you are using a CSpline for a router. If you want a division without any CSpline distortion, just use a default (linear, lower left to upper right) CSpline.

Router: (a CSpline gadget) If this CSpline is not in use, the bevels will be a single "ring" of polyspolygons. This CSpline causes the bevels to be composed of a number of rings set in "Bevel Segments", described above. Be careful! These can dramatically increase the number of polygons in the drawing.

The CSpline controls the shape of the bevel itself, very much like a router cutting along the edge of the poly. So, the "bevel" shape need not be a straight line, but can be a cove, round-over, or any of the complicated squiggles you could get from using a router on an edge.

If Bevel Segments is 1, the effect may be lost, depending on the position of the ends of the CSpline.

If the right end of the CSpline is back at "0", the bevel faces will be at a different position than seen in the interactive display.

The gadget brings up Spline Selection requester listing available CSplines. You can either select one that already exists or make a new one, clone an existing one, etc. A CSpline can be used to control the bevel in very complex ways that otherwise would require several scaling, stretching, shearing, and tapering operations.

As with other Aladdin 4D requesters, Bevel permits saving the parameters you set up, for later re-use of the menu item. The Bevel button performs the action you've defined. If you don't click save, the defaults will come back without your modifications next time.

Limits: Doesn't operate on special objects.

Edit / Advanced Tools Menu Item: Bridge
Keyboard: None

Brief: Adds "Skin" across selected points and polyspolygons.

Bridge lets you skin a "skeleton" of the points and polygons you select. The option opens the Bridge Settings requester.

Points Per Polygon: (default: -1) The value you enter here is the number of points you must select before a new poly is created.

The Bridge tool has a mode that allows variable number of points in the poly(s) it creates. If you use a value of -1 here, the tool will allow you to create polyspolygons of varying numbers of points. To use it in this manner, select the points, then move the pointer to a position on screen where there are NO POINTS and double click to create the poly.

The option will not recognize points of some special objects. When using Bridge, you should NOT have objects selected. If you do, it will not interfere with the operation of the bridge tool, though screen updates are not clear.

If you have the points per poly value set to 3, choose ANY 3 points from the existing ~~polys~~polygons~~.~~. As you select them, you will see a small box around the selected point~~.~~. When you select the third point, the new polygon (in this case, a triangle) is created and the points are unselected~~.~~. You may continue selecting points, creating new ~~polys~~polygons, or you may use <Esc> or the set command to exit the tool.

The points in the new poly will be created in the order that you have chosen them.

If you choose a point accidentally, you may "unchoose" it by holding down the <Shift> key and selecting it.

You can use the bridge tool to manually "face" a poly, just like the Face option~~.~~. To do this, it is usually convenient to select only the poly you're going to face~~.~~. This causes the poly to show its segments in alternating red/white segments, allowing you to easily see the segment endpoints.

You can also use Bridge to manually create a "bridge" between two existing objects~~.~~. For instance, if you had a sphere and moved the top half away from the bottom half, you could then use the Bridge tool to manually create a "tube" between the two halves~~.~~. In general, it is easier to see where the new ~~polys~~polygons should go if you hide all ~~polys~~polygons except those you intend to use~~.~~. When working in this manner, it is generally best not to have any ~~polys~~polygons selected before you invoke Bridge.

Some limitations apply to changing the view while Bridge is running, but you can still use the number pad keys to rotate your view~~.~~. When using the number pad to rotate the view, the synchronization of the display refresh might be delayed, so if they get out of sync, just tap one of the number pad keys (the <8> key is convenient) to cause the program to update the display.

As with other Aladdin 4D requesters, Bridge permits saving the parameters you set up, for later re-use of the menu item~~.~~. The Bridge button performs the action you've defined~~.~~. If you don't click save, the defaults will come back without your modifications next time.

Edit / Advanced Tools Menu Item: Conform
Keyboard: None

Brief: Conform selected ~~polys~~polygons to a spline.

This tool allows you to conform all selected polygons to a previously drawn spline~~.~~. The option brings up the Conform Settings requester.

Scale Width / Height: These two gadgets allow you to enter percentage scaling values.

Power: This function influences the "roundness" of the ~~polys~~polygons as the distance of the point increases from the plane of the center or Attach Point as specified.

Scale From: This toggle (Center or ATP) allows you to apply the Power drop-off from the center of all selected ~~polys~~polygons or from the current Attach Point.

Scale To: Another toggle ~~.~~ Near or Far~~Center or ATP.~~. In circumstances where the spline has two (or more) edges in line between the "from" point and the moving point, this allows you to choose which part of the spline to move toward.

~~/// These are Center / ATP ??? 0422 /j~~
~~@~~ If you choose far~~///FAR~~, the point will

@ converge on the segment of the spline farthest from the point. If you
@ choose //near, the points will converge on the segment of the spline nearest
@ the point.

Radial: Off or On. This fundamentally changes the amount of movement of the points related to their distance from the FROM point. If Radial is OFF, the points nearest the FROM point move the most. If Radial is ON, these points move the least.

Conform looks at the spline you have chosen and decides whether it has been drawn in the XY, XZ, or YZ plane. If it has been drawn in the XZ plane (usual in Y Flat View), the Conform option will then converge the points of the polyspolygons outward toward the spline shape in the XZ plane that the point is in when you start. The same reasoning applies if the spline is drawn in the other two axes.

To use Conform, draw a spline you want conformance to, just as though you were drawing a cross section of a shape. Place the spline around the polyspolygons you wish to have conform to the shape. Select the SPLINE first, then select any polyspolygons you wish to converge. Then select the menu option. The settings in the Conform Settings requester are automatically applied for normal invocation of the Conform tool in the tools windows. You can also use the tool in interactive mode.

Conform scaling is like that in the Scale option, but it is modified by the spline. If you don't like the results, hit <Esc> before setting the polyspolygons. If you like the results, use the Set command (RMB in the Editor window) to make the changes permanent.

Conform changes non-triangles to be severely non-planar. After using the conform tool you should almost always change the polyspolygons you have conformed into triangles before rendering, to avoid non-planar rendering artifacts.

Limits: Only works with polygons. Some special polyspolygons (like Flares and Fountains) are not affected.

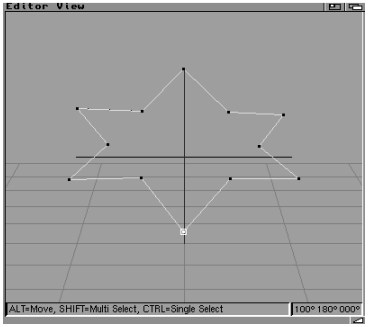
Tip: Conform converges the points of the selected polyspolygons as though they were moving outward toward a cylinder whose cross section is shaped like the spline. This means that if you anticipate the direction of the points in your original object, you can achieve a much more natural appearance in the finished conformed object.

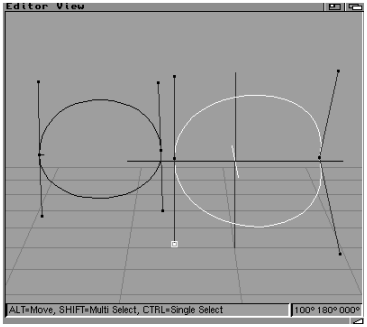
To illustrate this, lets say you use the Prim QPrimitive-Quad gadget to make a sphere. The poles of the sphere are aligned with the Z axis. Think of this as the axis of rotation of the object. The sphere is made with the Lathe tool, applied around the Z axis. Now lets say you draw a spline in the XZ plane, as is normal when in the Y Flat View, to conform the sphere to.

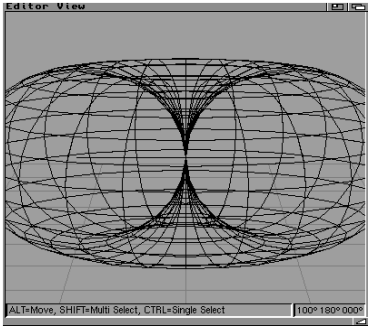
The "cylinder projection" of this spline is in the Y axis (the normal to the plane of the spline). The sphere, however, is aligned with the Z axis. You can, of course, use Conform this way, but the results are not as natural as when the sphere is aligned with the Y axis. Rotate the sphere 90 degrees on the X axis to align its poles with the Y axis. When you conform it to the spline, the conform is much more natural, resulting in far fewer polygons being non-planar. If you had drawn the spline in the XY plane (normal for Z flat view), the "cylinder projection" of the spline would be along the Z axis, and you would not rotate the sphere before doing the conform.

In these illustrations, you see a spline shaped like a six-pointed star and a sphere. The polyspolygons in either method should be converted to triangles before rendering; however, the bottom method will still give a much better result.

The radial option is demonstrated in the next three illustrations.

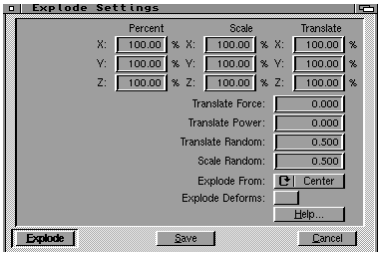






Here a torus was created with the Primitive-Quad ~~PRIM_Q~~ tool~~.~~. The torus has its "poles" aligned with the Z axis, just like the sphere, when it first appears~~.~~. Again, think of this as the axis of rotation of the torus, or the axis that would be chosen for the Lathe to create the torus.

Edit / Advanced Tools Menu Item: Explode
Keyboard: None



Brief: "Explode" selected ~~polys~~polygons.

Explode lets you blow up things~~.~~. That is, objects~~.~~. Turn them into parts that fly in all directions - or in the directions you choose, anyway~~.~~. Invoking the Explode menu item brings up the Explode Settings requester~~.~~. This requester contains entry boxes for Percent, Scale, and Translate for each of X, Y, and Z axes~~.~~. These let you enter percents for what you want to happen along what axis.

This option can be used with either entered values or as an interactive freehand exploder using the toolbox icon.

The exploded ~~polys~~polygons will change (either scale, translate, or both) in the two axes opposite to the selected axis at the top of the tool box, corresponding to the natural view direction when flat viewed.

If the Y axis is selected, (Y flat view) the ~~polys~~polygons change in the X-Z plane.

If the X axis is selected, (X flat view) the ~~polys~~polygons change in the Y-Z plane.

if the Z axis is selected, (Z flat view) the ~~polys~~polygons change in the X-Y plane.

Percent X, Y, and Z: These values are only used in PERFORM mode~~.~~. They are ignored when you use the toolbox to bring up Explode in freehand mode.

If the value is at 0.0, no change will occur~~.~~. A value of 1.0 exerts full change specified in the other defaults~~.~~. The value can be negative or positive.

In FREEHAND mode, these values are ignored~~.~~. Instead the values are based on the distance the mouse has moved and the qualifiers in effect.

Scale X, Y, and Z: These values indicate the desired amount to resize the selected ~~polys~~polygons~~.~~. The value is reached when the percent is at 1.0.

If the scale values are at 1.0, no resizing occurs~~.~~. If the value is at 0.5, the ~~polys~~polygons will resize to half when percent is 1.0~~.~~. If the value is 0.0, the ~~polys~~polygons will resize to nothing when percent is 1.0~~.~~. The values can be positive or negative.

Translate X, Y, and Z: These values indicate the desired amount the selected ~~polys~~polygons will move~~.~~. They are a multiplier for Translate Force.

If set to 1.0, the ~~polys~~polygons will move the full amount~~.~~. If set to 2.0, the ~~polys~~polygons will move twice the full amount~~.~~. If set to 0.0, the ~~polys~~polygons will not move~~.~~. If set to -1.0, the ~~polys~~polygons will move the full amount in the opposite direction~~.~~. Values can be positive or negative.

Translate Force: This value indicates the relative amount of movement that will occur~~.~~. Larger values increase this movement, and smaller values decrease it~~.~~. Values can be positive or negative~~.~~. A value of 0.0 gives no movement.

Translate Power: This value controls the relative velocity of the translation based on the proximity of the poly to the explosion center.

Say you have 2 ~~polys~~polygons, poly A, 10000 units from the explosion center and poly B, 20000 units from the explosion center.

If the value is 0.0, the ~~polys~~polygons will move at the same speed~~.~~. If the value is 1.0, A will move twice as fast as B If the value is 2.0, A will move four times as fast as B, if the value is -1.0, A will move half as fast as B.

Translate Random: This determines the amount of randomness to add the translation~~.~~. It is a multiplier for the force~~.~~. If it is set to 0.0, no randomness will be introduced~~.~~. If it is set to 0.5, the ~~polys~~polygons will move between full force and half force If it is set to 1.0, the ~~polys~~polygons will move between full force and no force~~.~~. If it is set to 2.0, the ~~polys~~polygons will move between full force and negative full force.

Scale Random: This determines the amount of randomness to add to the scaling~~.~~. It is a multiplier for the force~~.~~. If it's 0.0, no randomness will be introduced~~.~~. If 0.5, the ~~polys~~polygons will scale between full scale value and half scale value If 1.0, the ~~polys~~polygons will move between full scale value and no scale value~~.~~. If it's at 2.0, the ~~polys~~polygons will move between full scale value and negative full scale value.

Random Seed: This value controls the sequence of randomness generated. Each value creates a repeatable sequence. Values are whole numbers, 1 or greater.

Explode From: This ~~rotary gadget~~cycle gadget toggles where the Explode starts, either Center (the geometric center of the selected ~~polys~~polygons) or ATP~~.~~. the current Attach Point~~.~~. If the explode from is set to center, the changes will occur around the center of ALL the selected ~~polys~~polygons~~.~~. If it is set to ATP, the changes will occur around the current Attach Point.

Explode Deforms: Toggles between OFF and ON~~.~~. This lets you choose whether you want Explode to affect any deforms that may exist for the selected ~~polys~~polygons at the same time you Explode the ~~polys~~polygons~~.~~. THIS IS NOT UNDOABLE~~.~~. If the ~~polys~~polygons you change have deform levels, and if this option is ON, the deform levels will also be exploded, if OFF, they will not be affected.

~~/// is this removed ??? 0502 /j~~

@ RANDOM SEED: This value controls the sequence of randomness generated. Each value
@ creates a repeatable sequence.
@ Values are whole numbers, 1 or greater.

As with other Aladdin 4D requesters, this one permits setting up parameters for later re-use of the menu item. Explode executes the action you've defined. If you don't click save, the previous defaults will come back without your modifications next time.

Limits: Explode is meant to be used on polyspolygons. It ignores splines, Flares, Gases, Cameras, Targets, and Paths. It can be used on complex polyspolygons even if textured. Textured complex polyspolygons may have to have their textures reapplied, however.

Edit / Advanced Tools Menu Item: LOScut
Keyboard: None

Brief: Cut selected polyspolygons with a single poly along line of sight

LOScut is like a knife that you can use to chop holes in objects, and cut off parts of objects you either don't want, or want to make into independent objects. When you select LOScut from the menu, you should first have the polyspolygons you want to cut selected, and be careful to align the view so the cutter poly you want to use is lined up the way you want it to cut. Normally you'd use one of the flat views, but you may use any view angle. After selecting the tool, you'll be prompted to choose the poly that you want to cut with.

LOS means "line of sight," hence the importance of getting the view into position before you start. The LOScut option opens the LOSCut Settings requester.

Fill Holes: (Off or on) If your cutter poly is entirely contained within the poly(s) to be cut, you may choose that a poly be created to fill the hole created in the cut.

Delete Cut PolysPolygons: If ON, any polyspolygons that are cut will be deleted. If OFF, the original polyspolygons will still be there. They are hard to see since they exist in the same plane as the new polyspolygons created by the cut. Only original polyspolygons that are cut will be deleted.

Regroup In-Outside: If ON, any polyspolygons that are outside the cutter poly will be regrouped, as will any polyspolygons that are inside the cutter.

Intersect Only: If ON, a duplicate of the cut poly(s) will be made, but with points at the points of intersection of the cutter poly with the original poly.

Limits: PolysPolygons to be cut cannot be special polyspolygons (like Flares or Fountains), or have textures or shading applied.

Edit / Advanced Tools Menu Item: Point Control
Keyboard: None

Brief: Remove or add points in selected polyspolygons.

This tool allows you to increase or reduce the number of points in all selected polygons. It's controlled by the Point Control Settings requester.

Apply To: This ~~rotary gadget~~cycle gadget toggles between One and All~~.~~. It lets you choose whether the option works on one poly at a time, or on all selected ~~polys~~polygons.

Function: Here's where you choose whether to Add points or to Remove them.

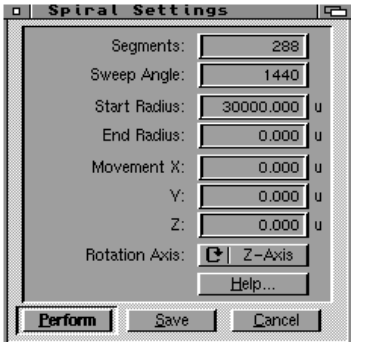
Direction: If Apply To is set to One, you are asked to choose a second point in the selected poly~~.~~. The operation is then applied between the initial point you selected the poly with, and the second point you choose. If Direction is set to Forward, the program operates on the segments of the poly between the first and second point in order~~.~~. If it is set to Backward, the program works on the segments in reverse order.

Angle: (useful only when removing points) This allows you to choose a minimum angle (in degrees) that must be encountered between segments before the point between the segments is removed.

Points: (only when adding points) This is the actual number of points that will be added in each existing segment.

To use the option in ONE mode, make sure you know which way the points are ordered, and whether you want the tool to work Forward or Reverse~~.~~. Choose the poly you wish to change by the First Point in the area you wish to change~~.~~. If when prompted to choose the second point, you choose the same point as the first, the entire polygon is done~~.~~. This action is NOT undoable, so you may want to clone the poly and hide it or jump it to a new space, so you can get it back if you need it.

Limits: Will not operate on most special ~~polys~~polygons (like Flares and Fountains).



Edit / Advanced Tools Menu Item: Spiral
Keyboard: None

Brief: Create spiral polygons.

This tool can be used to create single polygons that have a spiral shape~~.~~. These are useful for motion paths, or extrusion paths~~.~~. They are not generally used directly for rendering~~.~~. Selecting the option brings up the Spiral Settings requester.

Segments: This numeric entry box sets the number of segments (or points) in the spiral shaped poly.

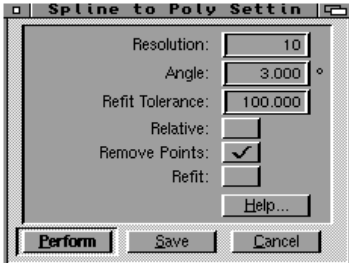
Sweep Angle: the angle in degrees that the poly will wrap when creating the spiral.

Start Radius: sets the beginning radius of the spiral.

End Radius: sets the ending radius of the spiral.

Movement X, Y, and Z: Allows translation along the given axes while the spiral is being generated.

Rotation Axis: (~~rotary gadget~~cycle gadget) choose the axis that the spiral will wrap around.



Edit / Advanced Tools Menu Item: Spline <-> Polygon
Keyboard: None

Brief: Convert selected splines to ~~polys~~polygons and ~~polys~~polygons to splines.

This option (also referred to as "SPOTOPOL") allows you to convert all selected polygons into splines, and all selected splines into polygons~~.~~. The option opens the Spline <-> Polygon Settings requester.

Resolution: (for spline to poly conversions) Here you set the number of points into which the program will divide each segment of the spline when constructing the polygon~~.~~. This will be modified by Relative if ON, and by Angle if Remove Points is ON.

Angle: (for spline to poly conversions) the minimum allowable angle (in degrees) between the sides of the poly created.

Refit Tolerance: If Refit is ON, then this number is the maximum distance that a point can be from a spline before it is used to modify the spline being created.

Relative: (spline to poly) If ON, the program will determine a smaller or larger resolution to use, based on the relative length of each spline segment~~.~~. The resolution you enter is still used as the base resolution, then a very short segment may get a smaller resolution, and a long segment will get a larger resolution.

Remove Points: (spline to poly) If ON, the program will remove points if the angle between the adjoining sides of the resulting poly is less than you specified in the Angle gadget.

Refit: (for poly to spline conversions) If ON, the program will treat the polygon as though its points were crude samples from a data set, and attempt to rebuild the data~~.~~. This results in the spline rounding out any angles.

The operation is performed on selected ~~polys~~polygons AND splines at the same time~~.~~. When the conversion is performed, the original spline or poly is not deleted.

When converting ~~polys~~polygons to splines, if Refit is OFF, the program makes a spline segment for each side of the polygon~~.~~. Control points are in place at one-third intervals for you to edit curvatures if desired~~.~~. However if Refit is ON, the program will treat the poly's points as though they were a rough part of a much larger data set and attempt to automatically fit a spline to these points.

The~~s~~re are some examples~~.~~ found in the external tools discussion of this same feature.~~.~~ ~~The polys~~polygons ~~are on the left, and the splines are on the right.~~

As with other Aladdin 4D requesters, this one permits saving the parameters you set up, for later re-use of the menu item~~.~~. Perform does the action you've defined~~.~~. If you don't click save, the previous defaults will come back without your modifications next time.

Limits: None.

~~/// NEED MORE HERE 0423 /j~~

Edit / Advanced Tools Menu Item: Wire Bend
Keyboard: None

~~/// ??? not included in external tools docs 0430 /j~~

@ Brief: -Bend selected polygons as in a wire jig. This option opens the Wire Bend Settings requester. It contains:

Restrictions: Doesn't operate on some special objects. Designed for use on polygons, however, it will alter the position of paths, lights, waves, fountains, flares, splines, and gases. (Note that some of these will have unusual results, as with the gases.)

Polygons are bent around a radius you specify, along an axis you specify.

If you think about a wire being bent around a cylinder, you can visualize the intended use for the tool.

This type of bend maintains the length of the wire at a constant.

If the circumference of the cylinder is 1 meter, a wire that is 1 meter long will wrap a full circle around the cylinder. In other words, as the wire is bent around the cylinder, assuming the wire is not stretched or compressed, it maintains dimensions along its length. Some part of the wire must be compressed and some part must be stretched to perform the bend. Perhaps the center of the wire remains constant in length, and the part closest to the inside of the bend compresses and the part closest to the outside of the bend stretches.

You have control over where the length is maintained by placing the attach point at this position.

IT IS STRONGLY SUGGESTED that you create a grid or gridded cube using the make rectangle tool to experiment with the tool when you are first becoming familiar. Use a 24x24 grid in the x and z axis, and use the bend tool with a Y rotation axis, changing the long axis from X to Z to see the difference. Change the attach point to different places on the grid and watch the difference. You may even want to place the attach point somewhere outside the grid to see how this alters the effect of the tool. Try different radius and angle settings. Use the measure gadget to see how it changes the radius to match the angle you request for the selected polygons.

DON'T FORGET that the use of point permission mode can greatly enhance this and other tools. Wirebend knows about the point permission mode and it is very useful when you want to only bend one end or side of an object.

For Entered Values:

1: Select the polygons you want to bend.
2: Click the RIGHT MOUSE BUTTON on the wirebend gadget.
3: Edit the values to reflect the bend you want
4: Select the Perform gadget.

The window will close and after the prep work, the polygons will be bent to the values and axes you have chosen. If you do not approve of the change, hit the escape key BEFORE using the set command. The polygons will be restored to their original positions.

If you hit the ACCEPT gadget, no change will occur, but the defaults you have entered will be there next time you open the tool in either freehand or entered value mode.

For Freehand (interactive) Bending:

1: use a RMB on the tool and set the options you desire, then ACCEPT the window.
3: Select the polygons you want to bend.
4: Click the LEFT MOUSE BUTTON on the wirebend gadget.

5: Move the mouse pointer into the editor window.
6: Press and hold the LEFT MOUSE BUTTON while dragging the mouse in a horizontal motion. You will see the selected polygons bending according to the amount you move the mouse and the settings you have entered in the control window.

The tool creates two circles centered to and at either side of the current attach point to help you visualize the bend radius in current use. NOTE that the tool enters quick mode for a faster redraw of the screen. These circles are actually polygons created just for the tool to use, and may not be painted if in quick move. You may want to turn OFF the auto quick (in the Settings menu) so the circles paint. This will slow down the editor update but may be desirable, especially when first becoming familar with the tool's operation.

The angle you have dragged the mouse to will be reported on screen. NOTICE that the angle will be automatically clamped to the maximum angle that the polygons can achieve. After this angle has been reached, moving the mouse further in that direction will not increase the angle. This limit is based on the length of the polygons along the Long axis at the current radius and is calculated by the tool during the prep phase. (For instance, a wire 1 meter long can only bend 180 degrees around a cylinder with a 2 meter circumference.)

7: Use the SET command (Right Mouse Button in editor window) when satisfied with the position of the polygons, or use the escape key to abort the operation.

The WireBend settings are as follows:

Radius: This is the radius that the polygons will be bent around, expressed in units. In the "wire bent around a cylinder" analogy, it is the radius of the cylinder.

Angle: This is the angle that the tool will bend the selected polygons. It is only used in the entered values mode. During interactive bending the angle is gotten from the mouse movement.

Measure: This gadget changes the radius so that the selected polygons will bend to the angle you have requested. NOTE that the polygons you want to bend MUST be selected (and the angle you want entered, and the long axis chosen) for the measure to work properly.

Rotation Axis: This is the axis that the polygons will be bent, or rotated around. Normally you would chose the axis that you "flat view" in to see the object in the way you want to bend it. For instance, a grid in the xz plane seems natural to bend around the Y axis, as it is naturally viewed. (You could, however, want to bend this grid into a cylinder, in which case you would use the x or z axis, and the flat view would be edge on to the grid.)

The long axis cannot be the same as the rotation axis.

Long Axis: Although the rotation axis tells the tool which axis to rotate the polygons around, it still needs to know which axis to use for the radius of the bend.

If you have a long cylinder that you want to bend, this is the axis that the length of the cylinder lies along. (Again, of course, you may want to "fan" the cylinder instead of bending it, so the other two axes may be chosen.)

Normally your intuitive use of the tool may be to choose the axis that the selected polygons have the longest dimension in, thus the name "Long" axis.

Note that the long axis is the axis that the measure gadget uses to establish the length of the selected polygons.

The rotation axis cannot be the same as the long axis.

Rotate From: If Center, the point of the bend where the distance remains constant will be the geometric center of the selected polygons.

If ATP, the point of the bend where the distance remains constant will be at the current attach point, giving you complete control over this point.

Note that the distance remains constant along a line along the long axis either at the center of the selected polygons or at the current attach point. During the rotation of the bend, points closer to the inside of the bend will be compressed, and points closer to the outside of the bend will be stretched.

~~@ Radius: 1.000 m/// NEED MORE HERE 0423 /j~~
~~@~~
~~@ Angle: 90.000 /// NEED MORE HERE 0423 /j~~
~~@~~
~~@ Measure (Button, checks objects selected)/// NEED MORE HERE 0423 /j~~
~~@~~
~~@ Rotation Axis (rotary gadget: Z, X, Y)/// NEED MORE HERE 0423 /j~~
~~@~~
~~@ Long Axis: (rotary gadget: X, Y, Z)/// NEED MORE HERE 0423 /j~~
~~@~~
~~@ Rotate From: (Rotary Button: Center, ATP)/// NEED MORE HERE 0423 /j~~
~~/// ??? need info on this one 0430 /j~~

Menu Item: Move to First
Keyboard: None

Brief: Put selected polygon(s) to first of list

This menu item reorders the selected polygon(s) to the first of the list in the current space~~.~~. If every time you click the wrong point selects, this option might help~~.~~. If the other point belongs to a polygon that was created first, it will always select, instead of the one you want~~.~~. If this occurs, you may be able to select the desired polygon at another location and move it to first~~.~~. Then it will always select first~~.~~. The real solution, however, is the next menu item, See Move To Last.

On file loading, the ~~polys~~polygons are always loaded in a specific manner: Paths first, then lights, then camera/targets, then gases, waves, and normal polygons~~.~~. This helps you select paths, camera/targets, etc., even when amassed in a group of polygons.

Limits: You cannot use this menu item during facing or deform editing~~.~~. Does not affect splines, Flares.

Menu Item: Move to Last
Keyboard: None

Brief: Put selected polygon(s) to last in list This menu item reorders the selected polygon(s) to the end of the list of polygons in the current space~~.~~. Often when you are trying to select a point, you find that it shares a location in space with another point~~.~~. If the other point belongs to a polygon that was created first, it will select, not the one you want~~.~~. If this occurs, you can simply select this menu item~~.~~. The

unwanted poly will move to the end of the list, and the desired poly will be selected before it.. See Move to First.

Limits: You cannot use this menu item during facing or deform editing.. Does not affect splines, Flares

Menu Item: Join ~~Polys~~Polygons
Keyboard: <F3>

Brief: Combine two polygons into one

This menu item joins any two polygons at the points you choose.. To use, select one of the polygons at the point you wish to join, then select this menu item, or press <F3>.. You will get a requester.. Click OK, then select the point on the other polygon that you wish to join.. The two original ~~polys~~polygons will be joined into a single polygon.. The order of the points in the original polygons determines how they are joined.. Often you will want to clone the originals and hide the clones.. Then join them.. If the join is not what you anticipated, delete the new poly.. Show the clones and reverse the order of one of them, then join them.

Joining ~~polys~~polygons is often to advantage in path design, and when you want to extrude two ~~polys~~polygons as one, as in the shape of the letter "O".

Limits: You cannot use this menu item during facing or deform editing.. The ~~polys~~polygons being joined must be free of shading and textures.. Does not operate on splines, Flares, or Fountains.. Some special ~~polys~~polygons are ignored.

Menu Item: Face (sub-menu)
Keyboard: (listed with sub-items)

Brief: Manual breakup of a poly.

This menu item allows you to manually break a complex polygon down into smaller ones.. There are four sub-items:

Begin: (<Ctrl> <F>) Select the polygon that you want to face, then select this sub-item.. All other polygons will be hidden, leaving you with an unobstructed view of the poly you are facing.

Add: (<F>) After beginning, you select the points of the polygon, one at a time.. The program will track which points you have selected in the order selected.. Each selected point will be darkened with a small black square.. When you have selected three or more points, select this sub-item.. You will see the new face appear.. Then continue with the additional faces you want.

End: (<Alt> <F>) When you have finished facing the poly, select this sub-item and program operation returns to normal.

Clear: (<Shift> <F>) Occasionally you will select the wrong point, or points in the wrong order.. Select this sub-item to clear the points previously selected.. This only clears the points, not previously created faces.

See also the external tool Breakup.

Limits: You cannot use this menu item during deform editing.. The poly being faced must be clear of shading and textures.. Does not work on splines, Flares, Fountains.. Some special ~~polys~~polygons are ignored.

A typical poly and the result of facing:

Menu Item: Match Points
Keyboard: None

Brief: Converge points within a specified range

This menu item finds points in the selected polygons that are very close, but not exactly the same, and moves one of the points to the other. This is important if you want to, say, move a hemisphere onto the end of a matching extruded pipe. If the points are even a single unit apart they will not shade properly.

With Match Points, you set an allowable distance before the match occurs. You must have the polygons selected. Unselected polygons are not examined.

Limits: You cannot use this menu item during facing or deform editing. Does not operate on splines and some special polygons.

Menu Item: Cleanup
Keyboard: None

Brief: Eliminate duplicated points

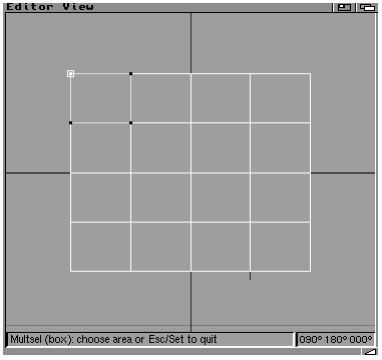
This menu item eliminates duplicated points in the selected polygons. To use, select the polygons you want to clean up, then select the menu item.

Limits: You cannot use this menu item during facing or deform editing. Polygons must be free of shading and textures. Does not operate on splines and some special polygons.

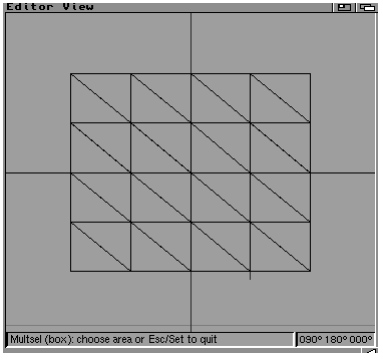
Menu Item: To Triangles
Keyboard: None

Brief: Convert selected rectangles to triangles

This menu item will find and convert any selected rectangles into triangles. This is important when you have altered the polygons until they are no longer planar or "flat". Polygons that are not flat will not render properly. In particular, textures, shading, and shadows cast from these polygons will become increasingly less accurate as the polygons become more warped in 3D. Shadows will show "holes" for polygons that are too far from being planar.



Aladdin 4D is somewhat forgiving about non-planar polygons in general. You may want to attempt your renders before using this tool, as it usually doubles the number of polygons in the object with a resulting increase in memory requirements and rendering time.



Limits: You cannot use this menu item during facing or deform editing. Polygons must be free of shading and textures. This

Tool only works on rectangles; complex polyspolygons must be faced.. See the Breakup external tool and facing.. Does not work on splines and some special polyspolygons.

Menu Item: Reverse Points
Keyboard: None

Brief: Reverse the point order in selected polygon

This menu item reverses the point order of the selected polygon.. This is useful when you want a path to move the polyspolygons the other way (does not affect rotation instructions, only movement), or when you're joining polyspolygons.

If you reverse the order of points on a textured or shaded polygon, the texture and shading will be changed and should be reapplied.

A related item is Choose First Point.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Choose First Point
Keyboard: None

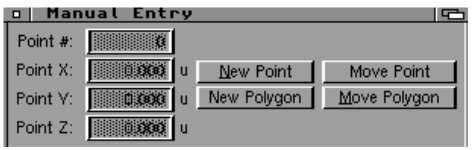
Brief: Allows you to reassign the First Point in a poly

This menu item allows you to change the defined First Point in a selected polygon.. To use, just select the polygon, then the menu item.. You are asked to OK the requester, then select the new first point.

You'd want to do this to define which point in a path that Aladdin 4D acts on for certain functions.. The First Point is the point that translates (movement) in the path during animation, for example.. This is also the point around which all rotation occurs around.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Manual Entry
Keyboard: <F10> ~~**(also right mouse button on FreeHand gadget)**~~



Brief: Specify exact coordinates for polygon or points

This menu item opens the Manual Entry window at the bottom of the view screen, and turns on Manual Entry mode.. This window is designed to work in conjunction with the Editor's normal operation, providing manual entries where mouse clicks wouldn't be close enough - or for any other reason.. You can still use the mouse to select, move, rotate, etc.. in the Editor Window.

When you select a polygon, (using the mouse as usual) you will see the 3D coordinates of the selected point appear in the Manual Entry window.. You can then edit them and click Move Point (keyboard: <m>) or the Move Poly (keyboard: <Shift> <M>).. You can cycle through the points of the poly by pressing the comma (<,>) or Less-Than (<) key and the period (<.>) or the Greater-Than (>) key.. There is a readout of which point in the polygon you're editing or have selected - indispensable for working with paths and specifying their First Points.

The Manual Entry window recognizes several keystrokes, if it's active. The windows are mutually exclusive. That is, only one window can be active at a time. So, if the Manual Entry window isn't active, the keyboard's input won't go there. Click anywhere in the Manual Entry Window to activate it, if you need the keyboard to work on it. A mouse click on its gadgets automatically activates the window without first clicking in it.

Here are the keyboard commands recognized by the Manual Entry window:

/// these keys do not work - are they to remain listed in the docs ??? 0423 /j

@ Key Purpose
@ ----------
@ <x> activate Point X gadget
@ <y> activate Point Y gadget
@ <z> activate Point Z gadget
@ <n> create new point (same as clicking the New Point gadget)
@ <Shift> <N> create new poly (same as New Polygon gadget)
@ <m> move current point (same as Move Point gadget)
@ <Shift> <M> move current poly(s) (same as Move Polygon gadget)
@ <d> delete current point
@ <Shift> <D> delete current poly
@ <F10> exit manual mode (same as Window Close)
@ <,> select previous point
@ <.> select next point
@ <s> execute the Set command
@ <1> select X active (<1> key top row, not number pad)
@ <2> select Y active (<2> key top row, not number pad)
@ <3> select Z active (<3> key top row, not number pad)
@ <Spacebar> Flat View in Active Axis
@ <q> toggle Quick Move
@ <Esc> UNDO (until actions are "set")
@ <Return> accept entry and deactivate gadget you are editing
@ <cursor keys> same as Editor: zoom, page move
@ <number pad> same as Editor: view angle change

It is possible to build a drawing one poly at a time using only Manual Entry mode, but normal operation is as a support and precision tool.

Limits: The program does not sleep during Manual Entry mode, so do not leave this window open if you don't need it. Although you can move a point (or control point) on a spline, you cannot build a spline from scratch using this tool, unless you build a poly and then convert it to a spline.

Menu Item: Trace Mode
Keyboard: None

Brief: Toggle trace mode

This menu item either allows you to load a bitmap for tracing or, if you have been tracing already, exits trace mode. To use it, just select the menu item - it's a toggle, remember. You will be presented with the file requester, with which to select a picture file to trace. The picture must be in one of the standard Amiga-displayable "ILBM" bitmap file formats. A racetrack window opens to indicate Aladdin 4D's progress in converting the picture file into a traceable image. When complete, the image appears as a background to the Editor's Window.

Before tracing, Flat View and record your view, so you can return to exactly this position if you change it for any reason. To trace, decide where on the image you want to begin a polygon. Enter FreeHand mode and draw a single straight line to this position. You will see the Editor immediately go to 2X magnification to help you trace. Set the new poly you've made.

This moves the Attach Point to the last position, and if you wish to make another poly, it will begin here. Click the FreeHand gadget again and draw the next desired poly. Continue until you have traced all the

polygons you need. You may find it easier to use splines in tracing curved areas. When using splines, the program does not go to 2X magnification.

When you're finished tracing, select Trace Mode again and the program resumes normal operation.

Limits: You must use a standard Amiga bitmapped image.

Menu Item: Track Attach Point
Keyboard: <Ctrl> <T>

Brief: Treat current Attach Point as origin

This menu item will allow you to keep a specific point in the drawing centered in the Editor Window. This helps in examination of a remote part of the drawing. You may want to use isometric mode when using this option if your drawing is too large. Also, all Page Move instructions are from the tracked point which is treated as the temporary origin.

Menu Item: Set Attach Point
Keyboard: None

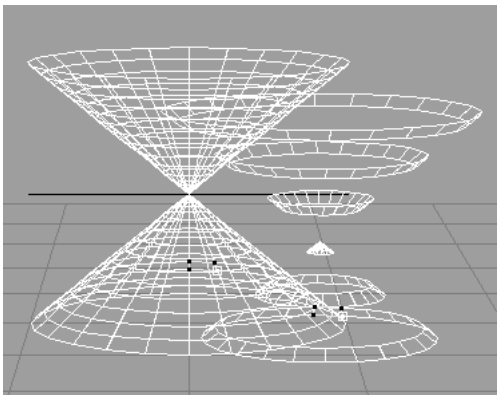
Brief: Specify new Attach Point location

This menu item allows you to manually enter a new location for the Attach Point. In practice, this is rarely necessary, but is of use in certain kinds of very precise drawing.

Limits: None

Menu Item: Set Quick
Keyboard: None

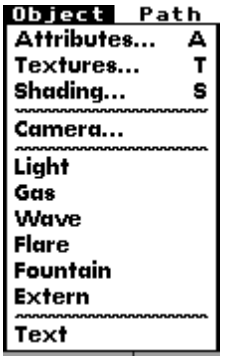
Brief: Issue Set command on Quick Move points only



This menu item can create some spectacular and bizarre effects that aren't possible any other way. Although rarely needed, the effect is startling. To use, create an object, perhaps using the Lathe tool. Select some point on the object. Tap the <q> key to enter Quick Move. Move the point you have selected before using the Set command, select the Set Quick menu item. This forces the ~~polys~~polygons to Set without applying the movement to the points that were not moved.

Limits: You cannot use this menu item during facing.

The Object Menu

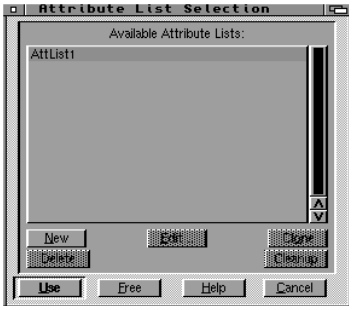


Menu Item: Attributes
Keyboard: <a>

Brief: Define, Load, Save, Edit, Apply Attribute Lists

This menu item is the entry point to the Attribute Lists. In rendering objects to the Amiga screens, Aladdin 4D consults its Attribute lists to find out what colors and other factors you've assigned to the surfaces for rendering. The Attribute List Selection requester lets you select lists from those that are available. You can also choose to create a new one, or edit an existing one. The "Clone" button will in addition permit copying an existing Attribute List for editing to your liking and application to the same or other objects or surfaces.

To select an Attribute List, click its name from the selection requester and choose an action from the gadgets. To use one, click Use, and it will be applied to the objects you've selected. To free one, of course, click Free.



A new drawing has no Attribute Lists. Use the New gadget to make one, and the Edit Attribute Settings requester will open, so you can define and name it. After you have an Attribute List, you can choose to edit it, apply it, or whatever.

Naming your Attribute Lists is a good habit to get into - and name them something meaningful, so you can tell what they apply to just from their names. When you re-use one for another purpose, if you think it'll get confusing to use it by one name, Clone it, and give it another one, to make your life as easy as possible.

The gadgets in the Attribute List selection requester:

New: Create a new list.

Edit: Open the Edit Attribute Lists requester. This requester is opened automatically if you choose New.

Delete: Delete selected list and unassign any ~~polys~~polygons that use the deleted list

Clone: Clone the selected list. No poly assignments are changed, but a new copy of the selected list is created, available for editing.

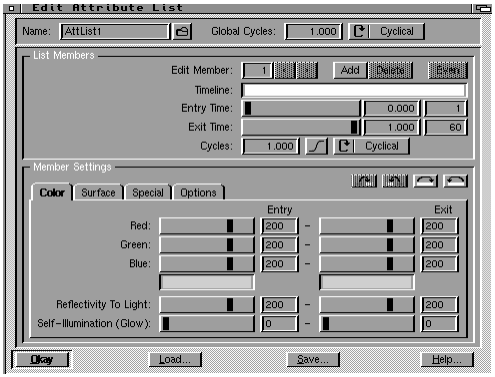
Use: If you choose Use, the changes you make are not recorded as new defaults for the option, but the changes take effect for the item you've been editing.

Free: Free the selected polygons of their Attribute List assignments

Cleanup: Go through all lists and if no polygons use it, delete it.

Accept: Close the window. Any polygons that are selected will be assigned the currently selected Attribute List.

The Edit Attribute Lists requester



This nearly screen-sized requester allows you to rename an Attribute list from its default or any other name.~~.~~ Attribute Lists may have the same name and still be different, but it won't take much of this to become confusing.~~.~~ You should name the lists something related to the ~~polys~~polygons you're defining them for, like "house.roof", or "Title: Hello".~~.~~ Names can have spaces, numbers, and extenders.~~.~~ In general try to limit the length of the name so that you can see the entire name in the selection window.~~.~~ When you go back to the selection window the new name will be shown and placed in order.

Load: Load an Attribute List from disk.~~.~~ The Amiga file requester opens, where you can pick the file you want.

Save: Save an Attribute List to disk.~~.~~ The file requester opens, where you can name the file.~~.~~ The name you save by doesn't have to match the names in the Attribute Lists.~~.~~ Aladdin 4D saves the entire list, not just the individual items in it.~~/// is this true ??? 0423 /j~~

List Members

Aladdin 4D's "Member" nomenclature permits assigning time slices of an animation to different Attribute (and other) Lists, even assigning them unequal portions of the animation.~~.~~ This control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters like Attribute Lists.~~.~~ The "Member" area of all of the requesters does the same thing, but applied to different properties.~~.~~ Initially, the requesters contain only one member, which is assigned the entire time line (see the graphic Time line gadget in the requesters).~~.~~ You
can add members, name them, and cycle through them to observe their whereabouts on the Time line.~~.~~ Each can have different specifications in the requester, and as you step through them, these specs will appear in the requesters' gadgets.

The Edit Attribute Lists requester is where all specifications are entered.~~.~~ Its gadgets and their use are:

Name: You can give your Attribute Lists names, and it's a good idea to do so.~~.~~ When you Save an Attribute List it is recommended that you use a meaningful name - like the name of the main list itself - followed by the extension ".atl", so you can recognize it as an Aladdin 4D Attribute List.~~.~~ It is also recommended that you save them to a single directory so they'll be easy to keep track of.~~.~~ None of this is required, however.

Global Cycles: The number of times the members of the list will be repeated during an animation

Cyclical/Periodic: If Cyclical, the members in the list will be used in an animation from start to end.~~.~~ If Periodic the members in the list will be used in an animation from start to end to start.

Member: Display the number of the currently selected member

Time line: an indicator gadget to graphically show the relationship of the currently selected member's time slice to the time of the animation.

Entry Time: (slider) Sets the beginning time for the member May be altered with the mouse.

Exit Time: (slider) Sets the ending time~~.~~. May be altered with the mouse.

Even: Force all members to an equal allocation of time.

Add: Add a new member to the list - it becomes selected.

Delete: Delete the currently selected member

Cycles: The number of times the current member's parameters will be executed during the member's time allocation

Member type: (~~rotary gadget~~cycle gadget) Cyclical, Periodic~~.~~. If Cyclic, the member's instructions will execute from start to finish~~.~~. If Periodic, the instructions will execute from start to finish and back to start.

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

The Color Tab:

Red, Green, and Blue Entry/Exit sliders: the amount of red, green, and blue - respectively - in the base color of assigned ~~polys~~polygons~~.~~. The color being defined is displayed below the sliders.

Reflectivity To Light: The amount of light that the assigned poly will reflect

Self Illumination (glow): The brightness of assigned poly with no incident light

The Surface Tab:

Hardness (Specular): The amount of specular reflection, or glossiness~~.~~. ~~Polys~~Polygons must be Phong shaded for this to apply~~.~~. In photography, a specular highlight is one that comes from a super-glossy surface, like a polished chrome object~~.~~. It contains no detail whatsoever, being entirely a perfectly white point of light~~.~~. The size of this point varies in nature with the size of the illuminating light source (usually the sun, a point source), and the shape of the object~~.~~. In 3D rendering, the term means much the same thing, except that specular reflectivity (that is, hardness) can be established for any surface to any degree desired~~.~~. Objects which in nature are hard, and reflective - billiard balls, metal surfaces, glossy plastics, etc. - will look best when modeled in Aladdin 4D if you apply Hardness (Specular) generously, according to the reflective nature of the surface being modeled.

Highlight Size (Gloss): The relative size of the specular highlights for assigned ~~polys~~polygons~~.~~. ~~Polys~~Polygons must be Phong shaded for this to apply.

Highlight Red, Green, and Blue sliders: The amount of red, green, and blue light - respectively - reflected by the surface~~.~~. ~~Polys~~Polygons must be Phong shaded for this to apply.

The color pots at the bottom of the tab show the color defined by the sliders.

The Special Tab:

Transparency: The amount of transparency for assigned ~~polys~~polygons

Trn Thickness: This adds opacity to assigned ~~polys~~polygons as they approach "edge on" in the view, simulating thickness to the transparent form~~.~~. Requires some transparency to be in effect.

Trn Light Sensitivity: This adds opacity to assigned ~~polys~~polygons as the amount of light on their surface increases~~.~~. Requires some transparency to be in effect

Wave Sensitivity: The "reflectivity" the assigned ~~polys~~polygons will have to waves sources in the drawing~~.~~. ~~Polys~~Polygons must be Phong shaded for this to apply.

Environment Reflectivity: The amount of reflection of Texture List members defined to be Reflection Maps~~.~~. The assigned ~~polys~~polygons must be Phong shaded for this to apply.

The Options Tab:

Enabled: Turns these options ON, when checked.

Background: This option makes ~~polys~~polygons behave as background ~~polys~~polygons~~.~~. This means that any ~~polys~~polygons that are farther from the viewer than the background poly(s) and covered by the background poly(s) will not be visible~~.~~. Instead, the currently defined background will appear~~.~~. Background ~~polys~~polygons are never visible~~.~~. You can use this to have an object "emerge" from some area in space~~.~~. Background ~~polys~~polygons may be a simple plane, or may be combined into other forms where objects can disappear, such as a cube in space.

Show:~~/// IS THIS REMOVED ??? 0505 /j~~
~~@ SHOW:~~ ~~Polys~~Polygons assigned to this member will be visible~~.~~. You can "turn off"
~~@~~ polygons with this switch for specific periods of time~~.~~. This flag operates
~~@~~ not only in rendering, but also in preview mode, so you can
~~@~~ verify the operation of the Show flag by previewing the animation~~.~~. You
~~@~~ should see the polygons appear and disappear as the corresponding members of
~~@~~ the Attribute List pass through their time lines.

Receive Shadows: ~~Polys~~Polygons assigned this attribute in a list may receive shadows~~.~~. ~~Polys~~Polygons must also be Phong shaded to receive shadows~~.~~. Also must have a local light with Shadows turned on and the Shadows global permission flag selected.

Cast Shadows: ~~Polys~~Polygons may cast shadows~~.~~. Also, you must have a local light with Shadows turned on and the Shadows global permission flag selected~~.~~. See Shadow Grouping and Render Settings for related information.

Self Shadows: ~~Polys~~Polygons may shadow other ~~polys~~polygons assigned to this member~~.~~. You should be knowledgeable before using this~~.~~. It can cause considerable screen artifacting~~.~~. ~~Polys~~Polygons must be Phong shaded for this to apply.

Samples: The small box with the number printed in it is the Shadow Acceleration parameter~~.~~. This is a "lossy" technique for ray tracing of shadows~~.~~. It is related in spirit to JPEG and other types of compression, but instead of compressing data, it compresses sampling with a corresponding decrease in calculation (and time) required~~.~~. Default value is 0~~.~~. Range is 1 to 255, with 1 being the slowest and most accurate, and 255 the fastest and most "lossy." A 0 setting means the option is not turned on~~.~~. The shadow compression is applied across each "span" - the horizontal area covered by a single polygon~~.~~. The acceleration parameter relates to the number of pixels covered by the span (though it is really not actually the number of pixels).

If the span is smaller than the requested acceleration, the acceleration is replaced by the span, to prevent extreme loss of shadow detail~~.~~. You don't need to understand all of this, just try using different numbers~~.~~. Keep your stop watch handy and you'll see the results~~.~~. As the value gets higher you may see a shadow extend into another shadow area, or a shadow area being missed completely~~.~~. This is normal for the

optimization and thus the "lossy" term~~.~~ Don't be too aggressive on this factor if you're rendering an animation~~.~~ Limit the numbers - not much acceleration, but not much "lossiness" either~~.~~ Of course for still frames you can always stop the render if you see a problem area, change the number, and render again.

An acceleration factor of 3 is approximately 3 times faster than normal ray traced shadows and at the worst case may miss a single pixel.

Line Type: This ~~rotary gadget~~cycle gadget displays Normal, Edges, Center, Points, or Point Center~~.~~ Clicking this gadget with the left mouse button will cycle Line Type through its available options:

Normal: The polygon will render in its normal state.

Edges: The polygon will render only along its edges.

Centers: The polygon will render without its edges.

Points: The polygon will render only at its points.

Point Center: The polygon will render without its points.

Width Entry/Exit~~// IS THIS REMOVED ??? 0505 /j~~
~~@ WIDTH ENTRY/EXIT~~: These gadgets allow you to specify the actual width of the
~~@~~ edges or radius of the points in units~~.~~ The entry and exit gadgets allow you
~~@~~ to change this over time during an animation so you can "fade" objects
~~@~~ in and~~/~~ out with Line Types.

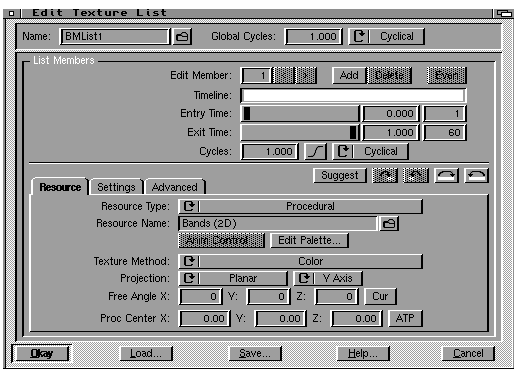
Line Types support all types of shading and texturing~~.~~ Try a simple sphere or cube and render an animation where the cube "fades" away using Edges with an Entry value large enough to paint the full polygons, and an Exit value of 0~~.~~ Complex objects when rendered with Points resemble the old surrealist paintings.

Also, Edges and Centers; and Points and Point Center are exact opposites, so you could, if desired, have two cubes in exactly the same space, one blue and one red, with opposite line types~~.~~ This will give a single cube with the corresponding area color which you can change smoothly over time.

The most obvious use of Line Types is for "girder" objects~~.~~ It's a snap to create girder tubes - for example, for the arms of your space station~~.~~ Another interesting use is in extruding fonts.

NOTE: shadows do not recognize the line types.

Limits: You cannot use this menu item during facing or deform editing~~.~~ Some definition options require Phong shading or special texture settings~~.~~ Some special ~~poly~~polygons do not use Attribute Lists.



Menu Item: Textures
Keyboard: <T>

Brief: Define, Load, Save, Apply Texture List

This menu item is the entry point to the Texture Lists~~.~~ When you invoke this item, it opens the Texture List Selection requester, in which you can select a Texture

List from the named ones listed in the requester and apply it to the object(s) that are highlighted when you invoked the menu item.. You can also use this menu item to Clone or Free a Texture List.

It's a good idea to give your Texture Lists meaningful names, so you can sort out what they do.. So, even though you might be using substantially the same settings for textures on two different objects, you should give them different names - Aladdin 4D allows duplicate names in the Texture Lists.

/// is the Cleanup option deleted ??? 0501 /j

To Edit an existing list, or to create a new one, click New in the Texture List Selection requester.

The Edit Texture List requester is where you specify texture qualities, choose bitmap or procedural textures to be applied, and establish the many parameters of doing so.. Of course, a new project has no Texture Lists, so you'll have to click the New gadget to make one.. The Edit requester is automatically invoked, where you can (re)name and define it.. Then after accepting the requesters, any polygons that are selected will be assigned the list.. You can assign it to other polygons by selecting them, and then invoking Texture from the menu.. Then choose the list you've just made - or any other.

When you select a Texture List, the program quickly looks at all polyspolygons in all spaces that are selected, to determine whether it will be permissible to add/delete members, alter mapping, etc.. Inappropriate gadgets will be disabled.. This prevents you from adding or deleting levels and changing illegal parameters when all polyspolygons that use a particular list are not selected.. When you clone a list, the Disabled flag is still set.. Select the original list you cloned, then Clone, to enable the cloned list.

The gadgets in the Edit Texture List requester are:

Name: Name: Display name and permit changing it Show names for selection (LMB).. Multiple Texture Lists can have the same name, but it will become confusing.. Name the list something related to the polyspolygons that you are defining it for - like "house.roof".. Names can contain spaces, numbers, extenders, etc.. In general, keep the length of the name so that you can see the entire name in the selection window.. When you go back to the selection window the new name will be shown and placed in order.

List Members

Aladdin 4D's "Member" section in its requesters permits assigning time slices of an animation to different textures and other elements, even assigning them unequal portions of the animation.. This control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters.. The "Member" area of all of the requester does this in a linear manner.

Initially, the requester contains only one member, which is assigned the entire time line (see the graphic Time line gadget in the requester).. You can add members, name them, and cycle through them to observe their whereabouts on the Time line.. Each can have different specifications in the requester, and as you step through them, these specs will appear in the requesters' gadgets.

Time line (bar): A visual representation of when the selected member will be active.. Unlike most other time lines in Aladdin 4D, Texture List members MAY exist at the same time.. This means you can blend textures.. Think of textures as layers of paint, laid down in order, first member to last member, with some partly or almost fully transparent.. If you have a later member that is at full strength and contributes color, it will cover any previous members that contribute color, no matter what their strength is.. Bump maps, however, divert the polygon's normal, so they may be full strength no matter where they appear and still have full effect.. Other types, such as reflection, illumination, etc.. may behave differently.. See the Briefs on each.

Edit Member: Changes the member you're working on.

Add: Makes a new member

Delete: Removes the currently selected member

Even: Causes an even distribution of time between all members

Entry Time: (slider) Sets the beginning time for the member May be altered with the mouse.

Exit Time: (slider) Sets the ending time. May be altered with the mouse.

Cycles: The number of times the current member's parameters will be executed during the member's time allocation.

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

Member type: (rotary gadgetcycle gadget) Cyclical, Periodic. If Cyclic, the member's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

Below the time line section of the requester are three tabs, Resource, Settings, and Advanced. These organize the texture's parameters into logical groups for you.

The Resource Tab

The Resource tab defines the texture itself - what type it is, how it behaves, and the external file that contains its bitmapped information or procedural instructions.

The available specifications can be given in this tab for each of Entry and Exit for animation.

Resource Type: (Rotary gadgetCycle gadget) If Bitmap, the texture will expect to load a bitmapped Amiga Iff picture of some kind for application to the surfaces of the polygons. Any Amiga IFF interleaved bitmap is legal, but of course you'll get the best results from 24-bit IFF images. You an use any software you like to create these bitmaps, including Aladdin 4D itself, but they must exist on disk before you can use them. If the surfaces you're texturing won't come too close to the camera, you can probably get by with using a lower-resolution, smaller-palette bitmap for its texture. Note the discussion of antialiasing, convolve, and other tricks for smoothing low-res textures, saving on render time.

If this reads "Procedural" - the default - Aladdin 4D is expecting a programmatic texture. These are usually much quicker to render, as they're programming instructions to the renderer, rather than external bitmaps. Many of these are provided with the program.. discussed at the end of this section.

Resource Name: This is the name of the bitmap or procedural texture you've chosen. It's informational only, but the gadget to its right gives you access to the file requester, if you decide to change it.

Anim Control: This selector lets you choose an animated sequence to be applied as a texture. It is for bitmaps only, and the bitmaps must be all of exactly the same size and resolution. Animation files won't work. You must have individual frames of animation - like the ones Aladdin 4D generates. Animation sequences must be all of the same name with a numeric filename extension in numerically consecutive order. The program recognizes extensions like .1, .2, .3, etc.; .01, .02, .03, etc.; .001,.002, .003, etc. or similar. Frames may be standard IFF, including 24-bit, or they may be DCTV format, JPEG compressed, or VideoToaster framestores.

To get Anim Control to work, you must load the first one of the animation frames in the sequence. When its name appears in the Resource Name box, you can continue. Clicking the Anim Control gadget opens the Anim BMR (Bitmap Resource Control) requester. Here you see the number of frames the program has detected by scanning through the available filenames with the same ste, name and a numeric extension. During rendering of the animation, Aladdin 4D will use the next frame of your selected sequence as it moves through the frames of the animation being rendered, according to the directions you give it in this requester. Only one frame will be loaded into memory at a time. The gadgets are:

Frames: Displays the number of frames the program has detected in the sequence.

Time: These gadgets allow you to specify when the animated texture should begin and end its play. This is represented as a percentage of the MEMBER'S allocated time. Time before the sequence begins will use the first frame in the sequence. Time after the exit time will use the last frame in the sequence.

Frame: These gadgets allow you to specify when the animated texture should begin and end its play. This is represented as a frame number. This is a percentage of the MEMBER'S allocated time. Time before the sequence begins will use the first frame in the sequence. Time after the exit time will use the last frame in the sequence.

CSpline (gadget): Allows you to choose a CSpline to control the rate and position in the sequence that is shown through the member's time

Cycles: This is the number of times the sequence should play.

CYCLIC/PERIODIC: If Cyclic, the sequence will play from entry to exit. If Periodic, the sequence will play from entry to exit and back to entry. You can reverse this or alter it in any way with the CSpline.

The changes you make to an animated bitmap sequence affect the sequence in ALL LISTS that refer to the sequence. If you want a different play rate for an animation you must rename the sequence and apply it as a separate bitmap texture.

You do not have to start with frame one of your animated texture sequence. When you load the animation sequence of single frames, choose a frame other than the first one in the sequence. The loader will then count this frame as the first and ignore any frames with lower numbers. You should make sure that when you reload a drawing using an animation sequence, ALL the frames used in the list are still present for the program to load when needed. If not, the program will re-count and use only those present - and if only one is present the animation instructions will be ignored.

Next to the Anim Control gadget is a gadget named Edit Palette. This gadget applies only if you've chosen a Procedural texture. It permits changing the palette which the Procedural texture will use. When you click this gadget, you open the Edit Procedural Colors screen. This is where you choose how many colors are to be used in the texture, the exact colors to be used, the spacing of the colors, and the amount of blend between them. The gadgets and their use are:

Colors: This is the number of colors (1 to 8) to be used by the Procedural texture.

Bell: If this is ON, the blend is performed according to an S-Curve algorithm, for rounder blends. If OFF, the blend is performed linearly. This is especially noticeable when procedural textures are used as bump maps, which are very sensitive to small changes in color.

Tuck: If this is ON, it changes all indexing into the color spacing by half of the first color. This prevents blending of the first color with the last color, especially useful in mirroring textures, like bands at the equator of a sphere.

L Blend and R Blend: The amount of blend between adjacent colors in the texture. This ranges from 0.0 to 1.0. You may specify individual left and right blend amounts for individual colors.

Set (for each of L Blend and R Blend: These gadgets set the blend for ALL colors in the selected area, Entry or Exit.

Entry: This graphic is where you select the entry colors. These colors are active at the beginning of the member's time period. When you select a color all gadgets in the window update to show you its values. During an animation the colors will shift from the Entry colors to the Exit colors. To change any color, click it and then operate the Red, Green, and Blue color sliders below.

Exit: This is where you select Exit colors, active at the end of the member's time period.

Start %: This gadget adjusts a color's spacing on the beginning side. The percents apply to fractions of the total palette, shown as Entry: and Exit: color swatches. If you change the Start % value, you'll see the spacing bars change to reflect any change. You can also enter an exact value in the trailing gadget if you prefer.

End %: This gadget adjusts the currently selected color's spacing on the Exit color swatches. The spacing of the colors on the bar changes to reflect any change you make here. You can also enter an exact value.

Red: Here you can change the amount of red in the currently selected color. Range is 0 to 255. You can use the slider gadget, or just type a number in the box at the end.

Green: The amount of green in the currently selected color. Range is 0 to 255.

Blue: The amount of blue in the current color - range 0 to 255.

Copy: If you click this gadget, you can copy the currently selected color to another area in the Entry and Exit color selection area. You will be asked if you want to also copy the Blend information.

Swap: Click this gadget to swap the currently selected color with another area in the Entry and Exit color selection area.

Even: If you click this gadget, the color spacing for the currently chosen number of colors will be evenly distributed.

Spread: This gadget spreads the currently selected color to another color that you click in the same selection area. This makes even gradations between colors, easily.

MATCH: If you select this gadget, the two color areas will be set the same. If you currently have a color selected in the Entry area, the Exit area will be changed. If you currently have a color selected in the Exit area, the Entry area will be changed.

~~/// This is missing ??? 0505 /j~~

~~@ COMPENSATE: This is a qualifier to the way you adjust the color~~
~~@ spacing. If this is OFF, when you change a color's spacing, it will intrude~~
~~@ on the next color, or leave open spaces. If it is ON, when you change a~~

@ color's spacing, the other colors in the active area will also be changed to
@ compensate for the change.

In the Texture / Resource tab:

Texture Method: When you click this gadget it rotates through all of the choices supported for the current member. There are several possible types:

Color: Texture contributes color as though a part of the polygon's surface.

Reflection Map: Texture contributes color as though the texture exists between the viewer and the polygon and the polygon is a glossy surface. This option requires the polygons to be Phong shaded, and the global Phong permission gadget must be selected.

Genlock: Texture contributes color as though a part of the polygon's surface. Any place Color 0 in the texture touches the polygons, the polygon becomes invisible. This option requires the polygons to have some transparency in their Attribute List, and the global transparency permission gadget must be on.

Decal: (Bitmap only) Texture contributes color as though a part of the polygon's surface. Any place Color 0 in the texture touches the polygons, the previous color on the polygon's surface remains unchanged. This is not available in procedural textures because it is the natural operation mode for procedurals.

Bump Map: Texture diverts the polygon's Normal based on the change in luma in the texture. This occurs before light calculations take place, resulting in a bumpy appearance. This option requires the polygons to be Phong shaded and the global Phong permission gadget to be selected. This option is not designed to work with overlapping transparent polygons.

Opacity: Texture modifies the transparency of the polygons based on luma. Dark colors in the texture become very transparent. Light colors become very opaque. This option requires polygons to have some transparency in their Attribute List, and the global transparency permission gadget must be ON. Also, the transparency is a modification of the base transparency of the poly. For full effect of the texture, use a transparency of 1 in the Attribute List.

Hardness: Texture modifies the specular reflection, or hardness of the polygons. Polygons must be Phong shaded and the global Phong permission gadget must be selected. This is a subtle effect, and in general Highlight Size should be quite large for gross effect. Highlight Size defaults to 200. A setting of 0 gives you the smallest possible highlight size. A setting of 255 gives you complete coverage of highlight anywhere the object receives light. Typical useful range is 0 to 254. When loading old drawings from previous versions of Aladdin 4D, you'll need to increase Highlight Size. as the range for this parameter has been extended in newer programs so that much smaller highlight sizes could be achieved.

Illumination: Texture modifies the amount of light incident to the poly, based on the luma of the texture. The lighter the area of the texture, the more light is added to the polygon's surface. This is often used as a base texture, with other textures covering its color content.

Projection: This gadget rotates through all of the Map Types supported for the current member. There are several Map Types:

Cylindrical: The texture is applied to the selected ~~polys~~polygons as though it is a cylinder projection from the axis that is designated in the Axis gadget.

Spherical: The texture is applied to the selected ~~polys~~polygons as though it is a spherical projection from the axis designated in the Axis gadget.

Spherical Point: This is a point projection spherical map - the industry standard type of spherical wrap~~.~~. To use it, select Spherical Point as the Projection type in the Edit Texture List requester and turn ON the Tiles option.

Free Angle: The texture is applied to the selected ~~polys~~polygons as a (parallel) projection but along the line of sight~~.~~. Choose the line of sight BEFORE using the Textures menu item by rotating your view angle to what you want for the texture application~~.~~. Then select the CUR (current Attach Point) gadget next to the Free Angle X, Y, and Z readout gadgets~~.~~. The program will calculate the necessary angles for you.

Shingles: (Bitmap only) The texture is custom fitted to each of the selected ~~polys~~polygons~~.~~. This is particularly useful if you want the same texture on the face of each letter in a font, or if you want to play back an animation on each facet of a sphere, etc.

Planar: The texture is applied to the selected ~~polys~~polygons as though it is a parallel projection through space along the axis designated by the Axis gadget.

Axis: This gadget cycles through the axes~~.~~. The axis you choose determines the direction for projection and wrapping types.

Free Angle X, Y, and Z: These are readouts of the Free Angle relationship with the Y axis that the texture will be applied~~.~~. They are only used with the Free Angle type~~.~~. Normal use is to align the texture visually in the Editor, then open the texture requester and use the CUR gadget~~.~~. Each member can have and will remember its own Free Angle set of angles.

CUR ("Current Attach Point"): (Procedural only - or Bitmap with TILES on)~~.~~. Places coordinates of the current Attach Point in the X, Y, and Z specifiers for Free Angle~~.~~. Normally, you set the Attach Point before using the Textures menu item~~.~~. This gives you a quick visual method of positioning the procedural texture's center.

Proc~~.~~. Center: (Procedural only - or Bitmap with TILES on)~~.~~. This represents the position in space that the procedural texture is centered~~.~~. For noise based textures, the noise will mirror on each side of the center, so it is normally put at some distance (100000, 100000, 100000 is typical) from the polygons~~.~~. For wrapping and 3D textures, it controls the center of the wrap or position~~.~~. When using a helix texture on, for example, a sphere, you normally want the center at the center of the object being textured.

ATP (Attach Point): Copies coordinates of the current Attach Point in the X, Y, and Z specifiers for Proc~~.~~. Center~~.~~. Set the Attach Point before using the Textures menu item.

~~@~~ Proc~~.~~. Orientation: ~~/// is this removed ??? 0505 /j~~
~~@~~ (Procedural only - or Bitmap with TILES on)~~.~~. This gadget
~~@ a~~allows you to rotate the texture as a final step~~.~~. If you use projection on
~~@~~ Z, and enter "10" here, the texture will be rotated 10 degrees from the normal
~~@~~ orientation~~.~~. This is particularly useful when using procedural textures as
~~@~~ backgrounds~~.~~. This is ignored in wrapping types.

The Settings Tab

Color: The amount of color to contribute from the member~~.~~ If zero, texture is converted to a black and white image based on luma~~.~~ If set to Max, the exact colors in the texture are used~~.~~ When using a texture as a bump map, this control supplements the bump map with color, resulting in "rubbing" type images~~.~~ For a pure bump map, set this control to zero~~.~~ The same reasoning applies to illumination maps~~.~~ If you want to add some color from an illumination map, use the color slider, but for a pure illumination map, set this control to 0.

Strength: The amount of effect this texture is to have~~.~~ Generally, use a partial strength to reveal previous texturing~~.~~ If no previous textures exist, the base color of the poly is blended when the strength is less than 1.0.

Tile: (Bitmaps only) This gadget alters the fundamental way a bitmap texture is indexed to a polygon~~.~~ If this gadget is ON, the program will put multiple (or partial) images on a single polygon~~.~~ This is especially useful if you want to place multiple images on a ground plane~~.~~ Try drawing a floor tile pattern in your paint program, saving it and placing it on a large rectangle with the Tiles gadget turned on~~.~~ This option works well with Free Angle and can adapt to procedural orientation~~.~~ Also try wrapping on a simple cube~~.~~ Unlike the Color bitmap wrapping types, which require a boundary in the ~~polys~~polygons at the bitmap edge, this type can wrap any shape without worry whether the boundary in the ~~polys~~polygons matches the bitmap edge~~.~~ The procedural center is used as the upper left index of the bitmap.

Anti-Alias: (Bitmap only) Will convert to an analog method of color determination~~.~~ Effectively eliminates individual pixels when a bitmap texture on a polygon gets too close~~.~~ Use this when the images on your ~~polys~~polygons are going to be larger on screen than the original bitmap was intended to be.

Filter: (Bitmap only) Uses a small tent filter to blur bitmaps.

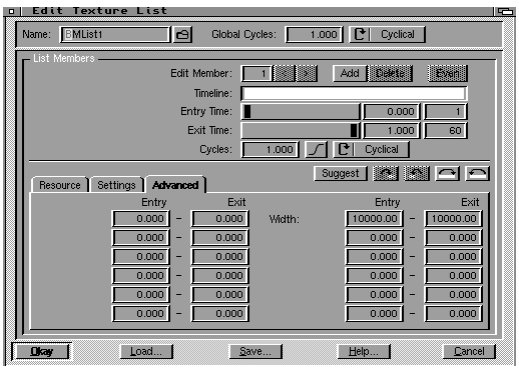
Negative: Turns the texture negative - black for white, for example~~.~~ Colors are also reversed to their complements.

X-FLIP: Reverses the X index for bitmaps and procedurals

Y-FLIP: Reverses the Y index for bitmaps and procedurals.

X-FLIP and Y-FLIP are quick ways to do the same operation that is possible with the X and Y indices~~.~~ Experienced users are encouraged to use the indices instead of these gadgets.

~~/// Gadgets in Advanced Tab have no names, just boxes of numbers 0505 /j~~



The Advanced tab

The gadgets in the Advanced tab will change based on the type of texture resource selected. Only gadgets with a label are used for the selected resource. Here is a complete list of the settings controlled by these gadgets:

First X Index: (Bitmap only) This value determines the first horizontal (X-axis) index into the bitmap for this member~~.~~ If the bitmap is 640 pixels wide and this is 0.0, the first pixel indexed will have an X coordinate of 0~~.~~ If this is 0.1, the first pixel indexed will be 64~~.~~ If this is 0.2, the first pixel indexed will be 128~~.~~ This makes it possible to "pan" and "flip" your bitmap textures during an animation~~.~~ There is no limit to the range of these variables~~.~~ You can use numbers smaller than 0.0 and larger than 1.0~~.~~ For instance, if

you use -1.0 for the First X ind and 2.0 for the Last X Ind, you will center the image, with equal space at the sides. This empty space is treated as though it is color 0 in the bitmap. The same reasoning applies to the following three variables.

Last X Index: (Bitmap only) This value determines the last horizontal index into the bitmap for this member.

First Y Index: (Bitmap only) This value determines the first vertical (Y) index into the bitmap for this member.

Last Y Index: (Bitmap only) This value determines the last vertical (Y) index into the bitmap for this member.

Width: (Procedural only - or Bitmap with Tiles on). Determines the distance in units that the procedural colors will span.

Height: (Procedural only - or Bitmap with Tiles on). Determines the distance in units before pattern repeats along the height.

Depth: (Procedural only) Determines the distance in units before pattern repeats along the depth.

Samples and Scaler@ SAMPLES and SCALER: A noise procedural is selected. the Samples is the
@ number of iterations the fractal noise generator will make. The Scaler is
@ both the divider for the next iteration and the amount of effect this
@ iteration will have. If you use one sample, the Scaler is not used at all.
@ Scaler values must be greater than 0.0 and less than 1.0. As you change one
@ value you may notice the other change. This is the program checking to see
@ if the combination requested will overflow doubles used in calculation, and
@ if so, resetting appropriately.
@
@ Remember the Scaler is a control over the "meander" factor of the noise. The best
@ range is 0.2 to 0.8. Samples is the number of times the Scaler will be
@ applied (plus 1). At 1, the Scaler is not used, at 2 it is used once, etc. The
@ larger the number of samples you use, the longer the render time. Samples is
@ rounded to a whole number even during animations, so a change in samples
@ will result in a sudden change from one frame to the next. Keep the entry
@ and exit samples the same while varying the Scaler freely. Try sample/Scaler
@ of 2.0/0.2, 3.0/0.4, 4.0/0.6 for good starting values.

@ PERSPECTIVEPerspective: A reflection type is chosen. This value controls the
@ overall amount of deflection of the image based on the angle of the polygons
@ reflecting the texture.
@
Scaler@ SCALER: A reflection type is chosen. This value determines how much of
@ the image is presented on the polygons. A large value will present multiple
@ images.
@
Screen Inf@ SCREEN INF: A reflection type is chosen. This value determines the
@ amount of influence the screen position of the polygon has on the portion of
@ the image used in the reflection.
@
Normal Inf@ NORMAL INF: A reflection type is chosen. This value determines the

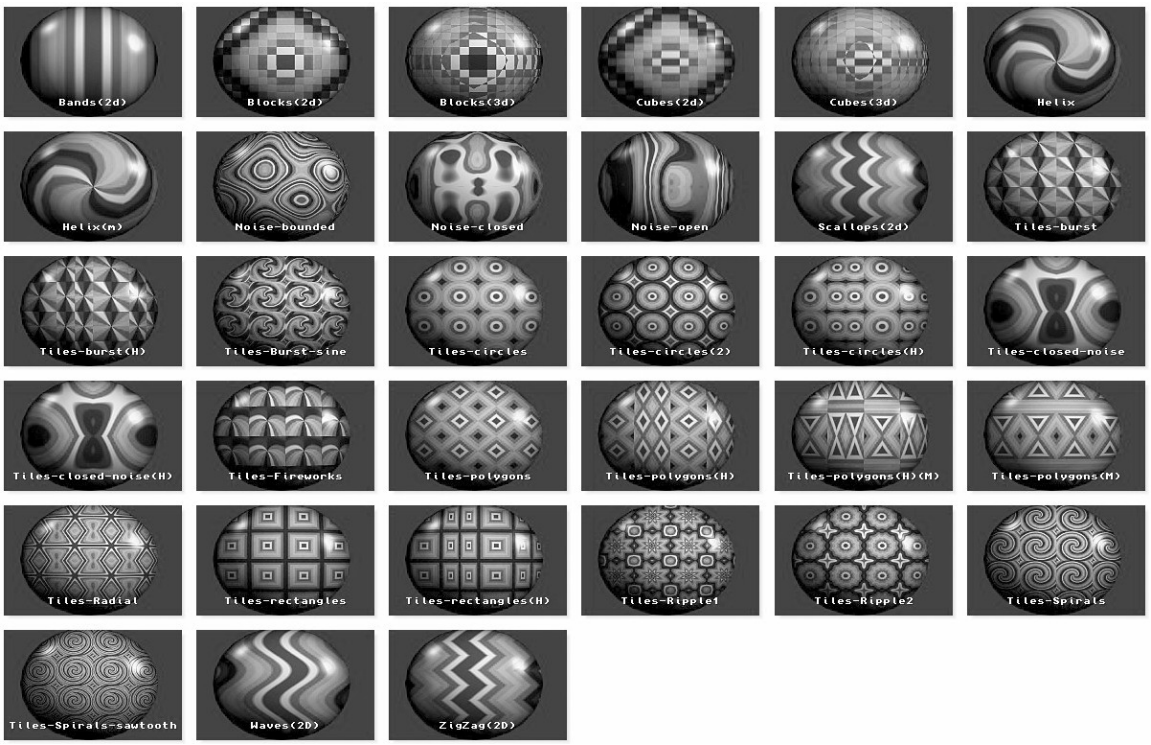
@ amount of influence the polygon's normal has on the portion of the image
@ used in the reflection.
@
@
@ NOISE WIDTHNoise Width: (Procedural only) The relative distance in units used to
@ scale the width of the noise aspects of the texture.
@
@ NOISE HEIGHTNoise Height: (Procedural only) The relative distance in units used to
@ scale the height of the noise aspects of the texture.
@
@ NOISE DEPTHNoise Depth: (Procedural only) The relative distance in units used to
@ scale the depth of the noise aspects of the texture.
@
@ AMPLITUDEAmplitude: (Procedural only) Waves, Scallops, or Zigzag is chosen. Value
@ is deflections distance along the width (x) of the wave, scallop or zigzag.
@
Burst@ BURST: (Procedural only) If Tiles: Burst is chosen. The value determines
@ how many times the procedural color radiate from the center of each tile. If
@ Helix texture is chosen, the value determines the number of times the
@ procedural colors are repeated around the axis that is being wrapped.
@
@ TwistTWIST: (Procedural only) A helix texture is chosen. This controls the
@ amount of twist introduced over the twist distance. 1.0 will twist the
@ colors once.
@
@ DistanceISTANCE: (Procedural only) A helix texture is chosen. This is the
@ distance over which the twist takes place.
@
@ Even OffsetVEN OFFSET: (Procedural only - or Bitmap with Tiles on). A tiles texture
@ is chosen. This is the amount of lateral displacement to be used on even
@ rows in the tiles. It is a percentage of Width.
@
@ Odd OffsetDD OFFSET: (Procedural only - or Bitmap with Tiles on). A tiles texture
@ is chosen. This is the amount of lateral displacement to be used on odd rows
@ in the tiles. It is a percentage of Width.
@
@ Cyl. BurstYL. BURST: (Procedural only - or Bitmap with Tiles on). For Wrapping
@ types this controls how many times the procedural pattern appears
@ during the wrap.
@
@ Sph. DistPH. DIST: (Procedural only - or Bitmap with Tiles on). A spherical wrap
@ is chosen. This determines the distance of the radius of the spherical wrap.
@
@ FrameRAME: (Procedural only) A tiles texture is chosen. If a value is here,
@ a frame will surround each tile. The value is a percentage of Width. Usable
@ range is 0.0 to 0.5. A value of 0.5 will completely fill the tile with the
@ frame color.

@ Blend DistLEND DIST: (Procedural only) A Tiles texture is chosen. This value
@ determines the distance over which the tile pattern will blend to the frame.
@ A frame width does not have to be specified. The value is a
@ percentage of the Width. Usable range is 0.0 to 0.5. The total of frame and
@ Blend Dist should be less than 0.5.

@
@ Range Scale ANGE SCALE: (Procedural only) A tiles texture is chosen. . This value
@ controls the number of times that the procedural colors are repeated from
@ the center of the tile. . Usable values are usually in the range of 1 to 5.
@
@ Clamp LAMP: (Procedural only) A tiles textures is chosen. . This value
@ determines the maximum value of the range scale that is repeated. . From this
@ point on, this value will fill the rectangle. . Usable values are 0.0 (no
@ clamp) to range scale.
@
@ # Of Sides OF SIDES: (Procedural only) A tiles: polygons texture is chosen. . This
@ value determines the number of sides used in the polygon internal to each
@ tile. . Usable range is 3 to about 12. . A value of 3 gives triangles; 4 gives
@ diamonds; 5 gives pentagons, etc.
@
@ Roll: OLL: (Procedural only) A noise-based texture is chosen. . This value
@ determines the "roll amount" through the noise. . It is particularly effective
@ to change this during an animation.

Included Procedural textures

The release version of Aladdin 4D includes 25 procedural textures. . You can use these as they are, or change them and composite them to suit your needs. . The included procedurals are:



Procedural Texture Descriptions

Bands (2D): Parallel bands of color

Blocks (2D): An infinite checkerboard

Blocks (3D): An infinite 3d checkerboard

Cubes (2D): Another checkerboard, but handles spacing changes differently than Blocks 2D

Cubes (3D): Another checkerboard, but handles spacing changes in a different manner than Blocks 3D

Helix: Twisting color bands, like a barber shop pole

Helix (M): Like Helix, but the twist reverses itself at the attach point

Noise Open: Fractal patterns that are all linked to each other

Noise Closed: Fractal patterns that are well contained

Noise Bounded: Fractal patterns that are well contained and regular

Scallops (2D): Sine type curves, but all positive, like the edge of a shell

Tiles Burst: Rectangular tiles with radiating color bars

Tiles Burst (H): Like Tiles Burst, but every other tile is split in two

Tiles Burst-Sine: A burst of color from center of each tile deflected by the sine of the distance from the center. The amount of deflection is Amplitude. Number of times deflection is used is Cycles

Tiles Circles: Rectangular tiles with circular patterns in them. The circles are based exclusively on width (always circular)

Tiles Circles (H): Like Tiles Circles, but every other tile is split

Tiles Circles2: Rectangular tiles with elliptical patterns in them. The circles are based on both width and height (ellipses if not equal).

Tiles Closed Noise: Rectangular tiles filled with the closed noise

Tiles Closed Noise (H): Like Tiles Closed Noise but alternates split

Tiles Fireworks: Burst of color from the center being deflected by distance from center. Amplitude is amount of deflection.

Tiles Polygons: Rectangular tiles with polygonal shaped color bars

Tiles Polygons (H): Like Tiles Polygons but alternates split

Tiles Polygons (M): Like Tiles Polygons, but alternate rows mirrored

Tiles Polygons (H)(M): Like Tiles Polygons (H) but alternate rows are mirrored

Tiles Radial: Burst of color from center with number of "arms" or sides. Use values of 0 and up for number of sides. Amplitude controls the force the colors are pulled up into the burst.

Tiles Rectangles: Rectangular tiles with rectangular color areas to match

Tiles Rectangles (H): Like tiles rectangles but alternates split

Tiles Spirals: Color burst twisted into a spiral pattern. Twist is number of times spiral should wrap.

Tiles Spirals-Sawtooth: Like Tiles Spirals, but spiral can be reversed on itself by adjusting the cycles.

Tiles Ripple1: "Daisy" patterns. Amplitude controls the amount of deflection, and Number of Sides controls the areas of deflection.

Tiles Ripple2: Like Tiles Ripple1, but amplitude is not modified by distance, resulting in a more mathematical, predictable pattern.

Waves (2D): Similar to Scallops, but both positive and negative parts of the sine wave. In Aladdin 4D 5.0, the Scallops 2D procedural

texture is now used (instead of Waves 2D) when chosen.

ZigZag (2D): A sawtooth pattern.

Limits: You cannot use this menu item during facing or deform editing. Some definition options require Phong shading or special Attribute List settings for the assigned polygons.

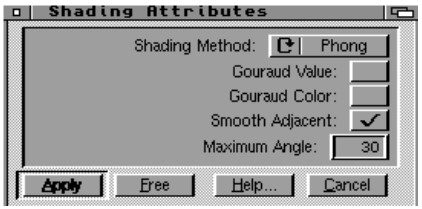
. Bitmaps can be loaded up to the memory limits of the computer, but enough memory must be left for rendering. Procedural textures take no additional memory, but the assigned polygons do require memory for indexing. If you are working in a limited amount of memory, keep checking the amount you have free, and save your work often.

Menu Item: Shading
Keyboard: <S>

Brief: Shade selected polygons

This menu item opens the shading assign window. This is where you tell the program how you want polygons to be shaded. Aladdin 4D can apply several different types of shading, which you can select for speed, realism, or to accommodate other features as needed for each object or polygon in a drawing.

Don't confuse "shading" with "shadows." Shading is the method by which the program calculates the rendered look of a surface. If you do not have any shading, you end up with a wireframe representation - probably not what you want in most cases. To apply shading, select the polygons you want to shade, then select Object / Shading. When the Shading Attributes requester opens, turn ON the options you want, and click Apply. You can use the same menu option to remove shading from an object by clicking Free, instead of Apply.



The gadgets in the window are:

Shading Method: (a ~~rotary gadget~~cycle gadget) offers None, Gouraud, or Phong shading types. These are discussed in excruciating detail in the Shading tutorial. Briefly, None means, of course, nothing. That is, no shading. Objects start life this way, unless you clone them from objects that have already had Shading applied. Gouraud calculates the incident light rays from any sources and figures out a color to make each polygon, as a whole. It looks "faceted."

Phong shading is the most thorough, and of course also the most calculation intensive. It looks at the light's incident vectors and shades polygons according to a variety of influences on their color, not necessarily resulting in a single color for the whole polygon. Some options require Phong shading, in order to work.

Gouraud Value: Use Gouraud shading on the value (luma) of the polygons.

Gouraud Color: Use Gouraud shading on the color of the polygons.
Smooth Adjacent: This is a qualifier to Phong shading. If on, it will create smooth (rounded) objects by averaging the normals of the adjacent polygons. If off, the polygons are still Phong shaded, which is necessary for some attribute and texture options, but the normal averaging is not performed.

This allows you to Phong shade, for example, a cube, even though you do not want the normals averaged.

Maximum Angle: Maximum Angle is the maximum angle permitted in the Smooth Adjacent operation. It is in degrees. You can select an entire object and shade the ~~polys~~polygons, automatically eliminating those ~~polys~~polygons that have angles greater than the one you specify. Of course, you can still shade manually for full control when needed.

There is no limit to the number of polygons that share a point in either Phong or Gouraud shading. Also, the program will NOT eliminate shading for polygons that have an angle of 90 degrees or less between them.

Limits: You cannot use this menu item during facing or deform editing. Gouraud and Phong shading are not permitted at the same time.

Menu Item: Camera

Keyboard: None

Brief: Create a new camera object or alter the current camera

When you select this menu item, it opens the Edit Camera And Targets requester. If a camera does not already exist in the current space, one is created, along with a single target. The target controls the camera's view direction and zoom. In this requester you specify how many targets to use and how each one is to control the camera.

When you use a camera, the Editor's view is ignored during preview and render. Instead, these draw the space from the camera's point of view. In the Editor you must keep all polygons in front of the view position. The camera, however, may move freely through the environment. Full 3D clipping is engaged, allowing polygons to move behind the camera position in part or in full.

In the Editor you see the environment and camera from the virtual viewpoint, and not through the camera. This allows you to select the camera and targets and move them around, assign them to paths, etc. The camera is represented on screen as a degenerate polygon - a type of polygon that is mathematically non-renderable and (should be) unique in your drawings - so that you can recognize it easily. It is also always visible no matter what axis or angle you're viewing from. The targets are shown as the same polygon, but are always half the size of the camera. They are numbered for easy identification. During preview you can see the targets, but they are not rendered in the final picture or animation.

Camera targets are time-lined. During animations the camera examines the active target to determine its zoom and view direction. If the target is not in transition between itself and the next target, the camera will look directly at the target. You can tell this in the preview because the target in control will remain directly in the center of the screen. If you have more than one target, and the target is in its transition time, the camera points at a position in space that linearly connects the target with the next target. This results in an extremely smooth and easily controlled virtual camera system.

The gadgets in the Edit Camera and Targets requester are:

Global Cycles: The number of times the instructions of the entire target system should be repeated.

Global Cycles Type (~~rotary gadget~~cycle gadget): Cyclical, Periodical) If Cyclic, the entire target system's instructions will be carried out from start to end. If Periodic, they will be carried out from start to end to start.

Near Plane: The front clipping plane of the camera. Normally this should be 0.

Far Plane: The back clipping plane of the camera. Polygons farther than this distance will be clipped.

Edit Member: Displays the number of the currently selected target. May be edited, or the <UP> <DOWN> gadgets may be used, or the time bar may be selected to change the currently selected target if there is more than one target in the current space.

Add: Creates a new target at the current Attach Point.

Delete: Remove the currently selected member

Even: Averages time over the current number of targets giving each an equal amount of time.

Time line (type in box): Name your time line here

Entry Time (slider): A visual representation of the start time of the target. May be moved with the mouse.

Exit Time (slider): A visual representation of the end time of the target. May be moved with the mouse.

Cycles (target): The number of times the target's instructions will be carried out.
CSpline: Allows use of a CSpline to alter rate of change between Entry/Exit values

(rotary gadgetcycle gadget) Cyclical, Periodic: The type of cycle you want. If Cyclic the targets instructions will move from beginning to end. If Periodic, the targets instructions will move from beginning to end to beginning. When using multiple targets, Cyclic should be used only if you are quite experienced.

Zoom (DW): Controls (Entry and Exit) the camera magnification. Normally this should not be less than 800, or the 3D clipping planes may be visible. (This can, however, be quite instructional.)

Tilt: The angle from vertical that the camera should be held in degrees, for each of Entry and Exit.

Transition: The percentage of the target's time that will be used to move to the next target. For instance, if the target is active from 0.0 to 0.5, and the transition is 0.5, the target will begin to pan to the next target at 0.25.

Transition Spline Gadget: Invokes CSpline control over the rate of change in transition.

Using Multiple Cameras To Achieve Transitions

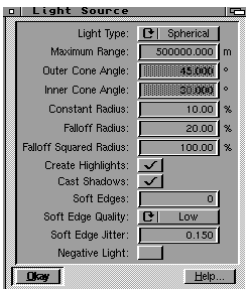
If you always use the camera with only one target, you can skip this part. You can have more than one camera, however, so Aladdin 4D's basic method of camera operation needs further explanation.

First, notice that the time line has the transit part of each member as a "grayed" area. The target instructions, zoom and tilt, are averaged, moving from the Entry values to the Exit values BEFORE the transition begins.

Also, the Cyclic and Periodic flags only apply to the pre-transit time. The transition only occurs once, regardless of the number of cycles. It can still, of course, be made Periodic if desired using a CSpline.

Limits: You cannot use this menu item during facing or deform editing. The camera may not use isometric mode. However, the camera is permitted to move into the interior of a gas container for some astounding effects.

Menu Item: Light (sub-menu: New, Edit)
Keyboard: (Keyboard: <Shift> <L>)



Brief: Create or edit a light source

This menu item allows you to either create a new local light, or edit an existing one. The sub-menu item New creates a new light at the current Attach Point and brings up the Light Source requester. Edit works on previously created lights, and brings up the same requester. One light must be selected before

selecting the Edit sub-menu item.. New lights are created at the current Attach Point.. Lights are represented in the Editor as cute sparkly stars so you can distinguish them easily.

The gadgets in the Light Source requester are:

Light Type (rotary gadgetcycle gadget): Spherical or Conical - This is the "type" of light source you want to create.. You can change this using the Edit sub-item if you need to.. Spherical light sources

Maximum Range: This is the radius of the effectiveness of the light.. It is expressed in units measured from the center of the light.

Outer Cone Angle: (enabled for Conical only): 45.000 is the default

Inner Cone Angle (enabled for Conical only): 30.000 is the default.

Constant Radius: This is a percentage (0.0 to 1.0) of the light that will be contributed and is uniform within the range of the light.

Falloff Radius: This is a percentage (0.0 to 1.0) of the light that will be contributed and will drop off linearly according to the distance from the light.. This isn't as natural as Falloff Squared Radius, but it's great for solving tricky light problems, and for lights that are too close to (or far away from the objects they illuminate for Falloff Squared Radius to produce the desired effect.

Falloff Squared Radius: This is a percentage (0.0 to 1.0) of the light that will be contributed and will drop off according to the square of the distance from the light.. In nature, light falls off in this manner ("inverse square law" to photographers).. Although the this option makes the drop off of light behave according to nature, most 3D works recommend using a percentage of the three types (see the default settings), instead.. You may want to experiment with these, to determine a "look" to your liking.

Create Highlights: If ON, the light will cause highlights on polygons that are Phong shaded and have some hardness.. If OFF, it will not.

Cast Shadows: (Check box) This yes/no choice tells the program whether the light source is to cast shadows.. Turning this on will probably result in additional rendering time.. However, you can leave it off during testing, and turn it on for the final render.. If ON, the light will cast shadows on polygons that are within the area its light falls on, and that have suitable Shadow attributes set.

Soft Edges: An integer from 0-255 which determines how soft to make the edges of shadows. A value of 0 disables soft edges entirely, and a value of 255 creates very soft edges. /// ??? need information /0425 /j Note that using soft edge shadows will dramatically increase rendering time.

Soft Edge Quality: (rotary gadgetcycle gadget)) The quality of the soft edges. Low quality is faster, but may produce undesirable artifacts. High quality produces the best possible edges, but takes longer to compute. This parameter is only used when the Soft Edges value is 1 or greater. Low
Medium
High

Soft Edge Jitter: Jitter controls the amount of “dithering” applied to the blended edges of soft shadows. A value of 0.000 performs no dithering, resulting in noticeable banding effects. In most cases, the default value of 0.150 is suitable. This parameter is only used when the Soft Edges value is 1 or greater. /// ??? need info 0425 /j

Negative Light: (Check box) This yes/no choice tells the lights to cast ‘darkness’ instead of illuminating an area. Excellent for controlling atmospheric lightning or putting that extra amount of darkness underneath an object in a scene.

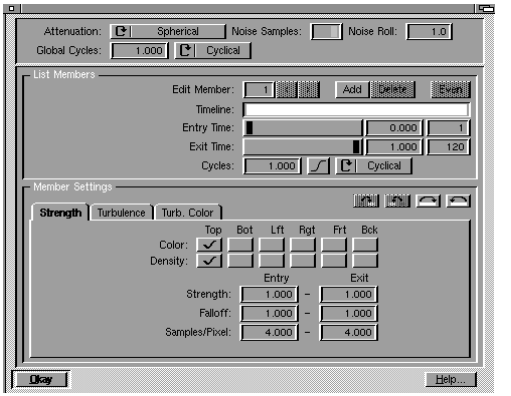
LIGHT COLOR AND STRENGTH:

A light gets its color and strength from an Attribute List. This means that you can take advantage of all of the abilities present in the list concept to alter the light's color and strength during an animation. The light must be assigned to an Attribute List to take advantage of this.

Four gadgets in the attributes affect a light. They are the R G B sliders, which determine the color of the light, and the reflectivity, which determines the strength of the light.

Limits: You cannot use this menu during facing or deform editing.

Gas



Menu Item: Gas (sub-menu: New, Edit)
Keyboard: (listed below)

Brief: Create or edit a gaseous object

Aladdin 4D allows you to define areas in space that can have density applied as though the space were in a fog - a gas, in other words. These are called gases or gaseous objects, or "procobjects" in Aladdin 4D - that is, special ones. Gases are volumes defined in a drawing where the space itself is modified, not just the surface of a polygon. See the tutorial on gases for further exploration of the principles.

The Object / Gas menu item allows you to either create a new gaseous object, or alter an existing one through its two sub-items. The New option, of course, makes a new one. The keyboard shortcut to make a new gas is <Shift> <L>). To alter an existing one, choose the Edit sub-menu. Its keyboard shortcut is <Alt> <L>). One Gas must be selected before invoking the Object / Gas / Edit menu item.

This menu item opens a requester where you can do just that. The controls are:

Attenuation: This ~~rotary gadget~~cycle gadget chooses the "shape" of gas to be created (or changed to, if you're editing an existing gas). The available shapes are: Spherical, Solid, Top-to-Bottom, Bottom-to-Top, Left-to-Right, Right-to-Left, Front-to-Back, and Back-to-Front. These types are pretty much described by their names...

The descriptions like "left" and "top" refer to the cubical container as viewed from the Y Active Axis. They are:

Spherical: The container holds a spherical (oblate) gas

Solid: The container holds a constant gas

Top to Bottom: The gas lessens in density from top to bottom

Bottom to Top: The gas lessens in density from bottom to top

Left to Right: The gas lessens in density from left to right

Right to Left: The gas lessens in density from right to left

Front to Back: The gas lessens in density from front to back

Back to Front: The gas lessens in density from back to front

Noise Samples: A subtle difference in the rendering of the Gas can be made by changing this variable. It's the number of times the fractal noise is used to create the turbulence. Normal values are 2 to 4.

Noise Roll: This "rolls" the turbulence. A value of 1.0 will cause the turbulence to roll once during the animation. A roll is an inversion in density for the turbulence at any particular point in space. If the density is 0.0 at a point in space when the animation begins, a roll of 1.0 will cause this point to have a density of 1.0 halfway through the animation and back to 0.0 at the end. The result mimics the rolling look common to fog or mist.

Global Cycles: The number of times the gas's instructions will be executed during the animation. This specifies a global value for the whole Gas. It's affected by the ~~rotary gadget~~cycle gadget "Cyclic, Periodic" which determines what kind of cycle the Gas (globally) is to have. If Cyclic, the gas's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

List Members: Aladdin 4D's "Member" nomenclature permits assigning time slices of an animation to different specifications for the Gas, even assigning them unequal portions of the animation. Control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters of the Gas. The "Member" area of all of the requesters does the same thing, but applies to different properties linearly, rather than in a mathematically calculated manner. Initially, the requester contains only one member, which is assigned the entire time line (see the graphic Time line gadget in the requester). You can add members, name them, and cycle through them to observe their whereabouts on the Time line. Each can have different specifications in the requester, and as you step through them, these specs will appear in the requesters' gadgets.

Edit Member: 1 (UP/DOWN gadgets step through the available Gas definition members)

Add: Makes a new member of the list

Delete: Gets rid of one.

Even: Averages the available time evenly between all members.

Time line: (text string box) Which one is in operation

Entry Time: (slider) Choose an Entry for the current member

Exit Time: (slider) Choose an Exit time for the currently selected member

Cycles: The number of times a member's instructions will be executed

Spline Gadget: Apply CSpline control to the Gas's behavior over time.

Type: (~~rotary gadget~~cycle gadget) Cyclic, Periodic..~~.~~. If Cyclic, the gas's instructions will execute from start to finish~~.~~. If Periodic, the instructions will execute from start to finish and back to start.

Member Settings: The specifications for a Gas member's behavior

The lower area of the requester is conveniently divided into tabbed areas, to organize the parameters logically.

Gas color and Density

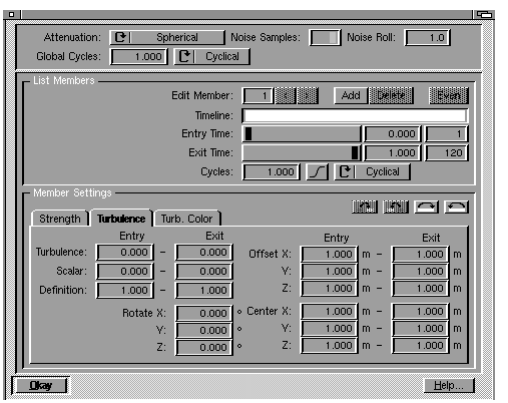
~~(Tab)~~ Strength Tab

The Strength tab is also where you set the color of the Gas~~.~~. Here you'll find gadgets with which to specify Color and Strength for each of the "corners" of the Gas: Top, Bot, Lft (or Left), Rgt (Right), Frt (Front), and Bck (Back)~~.~~. The gadgets represent where the gas will get its density~~.~~. This is treated as a modifier to the Strength setting you're using~~.~~. You can have any number of them selected, but you must always have at least one~~.~~. If you have Top selected, the gas will modify its density based on the transparency set for the top polygon of the cubical container~~.~~. Of course, the polygon gets its transparency information from an Attribute List~~.~~. This means the gas can change density during the animation not only from the Strength Entry and Exit values, but also from the Attribute List~~.~~. If you have Top and Bot both selected, the gas will modify its density from the top and bottom~~.~~. This is performed linearly based on the relative distance of the gas sample from these two polygons~~.~~. In other words, the gas can have a smooth blend in density between these colors~~.~~. For this, you would use separate Attribute Lists for the top and bottom polygons~~.~~. Up to six sides can be used with up to six different Attribute Lists.

Falloff: The power function that controls the density drop off in density for the attenuation type chosen

Samples Per Pixel: The number of samples of the gas that will be taken and averaged for each pixel that the gas container occupies on screen during the render~~.~~. The more samples you use, the longer the render time, but the smoother and more accurate the gas is represented~~.~~. Some very impressive images can be created with only one or two samples~~.~~. Normal images will use three to six~~.~~. Extreme accuracy is achieved with higher numbers~~.~~. No limits are placed on this value, except that it should always be positive.

~~(Tab)~~ Turbulence Tab



The Turbulence of a gas is specified as Entry and Exit values for each of:

Turbulence: If a value is present here, the gas will have a noise based fractal change in density.

Scalar: The size of the turbulence

Definition: The power function that controls density drop off for the turbulence

The source for the Turbulence is specified as Entry and Exit values for:

Rotate X, Y, and Z: The position in the respective axis of the center of rotation of turbulence.

Offset X, Y, and Z: The position in the respective axis of the center of turbulence at the start of the member.

Center X, Y, and Z: The center of the turbulence. /// ??? isn't this redundant to rotate and offset ??? 0425 /j

(Tab) Turb. Color Tab

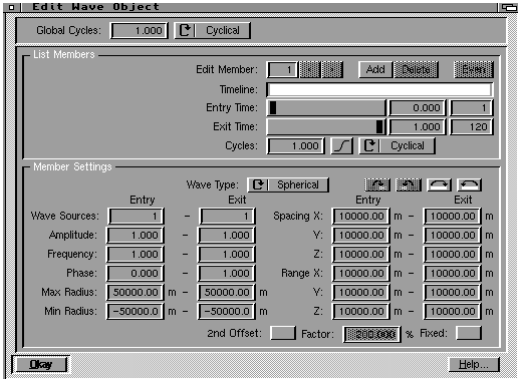
Turbulence can be a different color from the gas. You can have green turbulence in the yellowish atmosphere of your far-flung planets, for example. The Turb. Color tab is where you set up that second color. Here you can specify a color scheme for the turbulence that will help distinguish turbulence within the gas. The color is applied to the gas based on turbulence strength, which varies according to the strength of the fractal noise at any position in space. The visual result is that the turbulence is "edged" in a color of your choice.

The polygons that make up a Gas "container" can be assigned Texture Lists. If you use Texture Lists on the polyspolygons of the container, the density of the gas will be modified by the luma value of the textures. You can use procedural, bitmap, and even animated sequences. All controls for Strength and Color are honored. Procedurals are indexed normally, but bitmaps are always applied as projection types along the main axes. You can, however, use Flip and Start/End "pan" variables to control their changing locations.

Aladdin 4D permits the camera to enter a gaseous space, and will render glorious results. However the render times increase dramatically. Even so, the excellent look of this effect is often worth the extra rendering time.

Limits: You cannot use this menu item during facing or deform editing. Gases always maintain their original relationship to the world axes. If you change the position of a point in the container of the gas, you change the size of the gas container, and this will be shown in the Editor. Bitmaps are limited in the method of application for gases and always appear on the faces of the container as though projected along one of the main axes.

Menu Item: Wave
Keyboard: (at sub-items)



Brief: Create or edit a Mechanical wave

This menu item has two sub-menus, one for starting a New wave, and the other for editing an existing one. The keyboard shortcut for creating a New wave is <Shift> <W>. The keyboard equivalent of the Object / Wave / Edit menu sub-item is <Alt> <W>. One Wave Source must be selected before selecting this item.

Either way, the option opens the Edit Mechanical Wave requester, where Aladdin 4D lets you to define Wave sources in space. See the tutorial on waves. The gadgets in this window are:

Global Cycles: The number of times the wave is repeated (globally).

Type (rotary gadgetcycle gadget: Cylindrical, Periodic): If Cyclic, the wave's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

List Members

Aladdin 4D's "Member" section in its requesters permits assigning time slices of an animation to different Wave Sources and other elements, even assigning them unequal portions of the animation. This control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters. The "Member" area of all of the requester does the same thing, but in a linear manner. Initially, the requester contains only one member, which is assigned the entire time line (see the graphic Time line gadget in the requesters). You can add members, name them, and cycle through them to observe their whereabouts on the Time line. Each can have different specifications in the requester, and as you step through them, these specs will appear in the requesters' gadgets.

Edit Member: changes the member you're working on.

Add: Makes a new member.

Delete: Removes rid of the currently selected member.

Even: Causes an even distribution of time between all members.

Entry Time: (slider) Sets the beginning time for the member May be altered with the mouse.

Exit Time: (slider) Sets the ending time. May be altered with the mouse.

Cycles: The number of times the current member's parameters will be executed during the member's time allocation

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

Member type: (rotary gadgetcycle gadget) Cyclical, Periodic. If Cyclic, the member's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

Wave Member Settings

Wave Type (rotary gadgetcycle gadget): Clicking this gadget will rotate through the supported types of mechanical waves. They are:

Spherical: Waves will radiate in a spherical manner.

Linear X: Waves will radiate linearly along the X axis.

Linear Y: Waves will radiate linearly along the Y axis.

Linear Z: Waves will radiate linearly along the Z axis.

Multiple: Waves will radiate spherically, but space within the Wave Source's range will be treated as a three-dimensional grid, with the area within each cubical container in the grid treated as a separate Wave Source.

/// is Bumps removed ??? 0426 /j
@ Bumps: Wave Source is treated as a specific set of interacting Wave
@ Sources for convenience. This effect can be imitated by setting up

@ several Wave Sources using spherical types in proximity to each other.

Entry and Exit Settings for each of:

Wave Sources: If greater than one, the Wave Source will have an effect as though you've created this many independent Wave Sources all in very close, but random proximity causing subtle interactions in the wave patterns.

Amplitude: The amount of deflection of the normal of the polyspolygons in range. In other words, the Strength of the waves.

Frequency: The "closeness" of the waves.

Phase: Waves are basically sinusoidal functions, with low and high peaks that can move over time. Phase will cause the wave to move. Normal operation uses an Entry Phase of 0.0 and an Exit Phase of 1.0. This will cause the waves to move the distance on the polygon determined by their frequency toward the center of the wave. Values of 1.0 and 0.0 cause the wave to move outward from the center.

Max Radius: The maximum distance that the wave affects polygons sensitive to waves. Waves have two radii of influence. This allows you to create concentric "rings" of waves - like those made by a stone thrown in a pool of water. If you do not want "rings" set Min Radius to the negative value of Max Radius. If you want "rings" set the Min Radius to a value between Max Radius and the negative value of Max Radius. The closer the two values are, the smaller the ring of influence will be.

Min Radius: This is the minimum distance that the Wave affects polygons sensitive to waves. See Max Radius above.

Spacing X: The dimension of the grid in X.

Spacing Y: The dimension of the grid in Y.

Spacing Z: The dimension of the grid in Z.

Range X: The range of the sources in X.

Range Y: The range of the sources in Y.

Range Z: The range of the sources in Z.

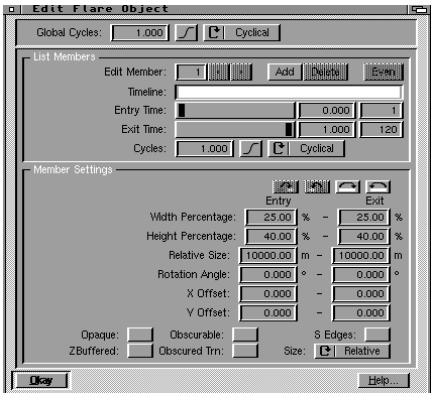
2nd Offset (check box): If checked, the program will create a secondary grid of waves whose size is controlled by the offset factor. This creates areas of interaction.

Factor: The relative size of the secondary grid to the primary grid. If value is 0.5, the secondary grid is half the spacing of the primary. If 2.0, it will be double. Non-integer (i.e. fractional) offsets, like 0.66666, or 1.33333 are normally used to emphasize interaction.

Fixed (check box): If this option is selected, polygons will react to a Wave Source at all times during the animation as though the Wave Source were moving along with them in their rotation and translation. If not selected, the waves move through and along the polygons.

Limits: This menu item may not be selected when in deform or facing modes.

Menu Item: Flare
Keyboard: (see sub-items)



Brief: Create or edit Lens Flare objects

This menu item allows you to either create a new lens flare, or edit an existing one, according to which sub-menu you choose. Flares require a Texture List to be assigned, and the texture that you assign must be a bitmap. Procedural textures won't work. After creating a Flare assign it a Texture List, and select a bitmap texture to apply to it. See the tutorial on lens flares.

In photography, flares are internal reflections, caused by the elements of a lens system. The more complex lens systems, such as video zooms and wide-angle still camera lenses, produce the spectacular lens flares when a strong light source, especially a point light source, appears in the lens' frame of view. Mathematically calculated lens flares, of course, do not respond to such lighting situations normally, since the math involved doesn't have much to do with actual pieces of glass. Lens flares are, in the photographic world, faults, and manufacturers go to great lengths to control them. A perfect camera lens, in fact, would generate no lens flares. The flares have, nonetheless, become an indicator of photographic "realism," so when panning across a huge spaceship, 3D animators drop in lens-like flares to suggest that a distant lens system is recording the scene.

In 3D rendering, flares are accomplished by a little more math - there's plenty to go around. The flare image, however, is calculated independently of the rendered image itself, and them composited onto the rendering as a last step. So, setting up a Flare is a matter of defining the flare itself, as well as setting up the parameters for the compositing step.

Object / Flare / New: (Keyboard: <Shift> <X>) Creates a new Flare at the current Attach Point.

Object / Flare / Edit: (Keyboard: <Alt> <X>) Edits a previously existing Flare. One Flare must be selected before invoking the menu sub-item. Flares are represented as triangles in the Editor Screen.

Both sub-items open the Edit Flare Object requester. The gadgets in this requester are:

Global Cycles: The number of times the Flare instructions are repeated during an animation

Spline Gadget: Turns control of the flare over to a CSpline

Cylindrical, Periodic (~~rotary gadget~~cycle gadget): If Cyclic, the Flare's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

List Members

The "Member" section of the Flare requester permits assigning time slices of an animation to different Flare definitions, even assigning them unequal portions of the animation. This control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters. The "Member" area does the same thing as the CSpline, but in a linear manner. Initially, the requester contains only one member, which is assigned the entire time line (see the graphic Time line gadget in the requester). You

can add members, name them, and cycle through them to observe their whereabouts on the Time line~~.~~ Each can have different specifications in the requester, and as you step through them, these specs will appear in the appropriate gadgets.

Edit Member: changes the member you're working on.

Add: Makes a new member

Delete: Removes rid of the currently selected member

Even: Causes an even distribution of time between all members

Time line: Manual entry box for the timing of a member - an exact begin and end time in percentage.

Entry Time: (slider) Sets the beginning time for the member~~.~~ May be altered with the mouse.

Exit Time: (slider) Sets the ending time~~.~~ May be altered with the mouse.

Cycles: The number of times the current member's parameters will be executed during the member's time allocation.

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

Member type: (~~rotary gadget~~cycle gadget) Cyclical, Periodic~~.~~ If Cyclic, the member's instructions will execute from start to finish~~.~~ If Periodic, the instructions will execute from start to finish and back to start.

Member Settings

Width Percentage: A multiplier for the width of the Flare on screen~~.~~ It is a percentage of the current render screen~~.~~ If the Size is fixed and if the current render screen is 640 pixels wide and the Width Percentage is 0.5, the Flare will be 320 pixels wide~~.~~ If, however, Size is set to Relative, the program will find the size of a line of Relative Distance in length at the Flare's position in space and then use Width Percentage as a percentage of this size so Flares reduce size as they become more distant.

Height Percentage: A multiplier for the height of the Flare on screen~~.~~ It is a percentage of the current render screen~~.~~ If the Size is fixed, if the current render screen is 400 pixels high and the Height Percentage is 0.5, the Flare will be 200 pixels high~~.~~ If, however, the Size is set to Relative, the program will find the size of a line of Relative Distance in length at the Flare's position in space and then use Height Percentage as a percentage of this size so Flares reduce size as they become more distant.

Relative Size: Flare size can be Relative or Fixed~~.~~ See the ~~rotary gadget~~cycle gadget to change this, if you need to~~.~~ If Fixed, the gadgets relate to the render screen~~.~~ If relative, the gadgets relate to the Relative Distance gadget entry as discussed there.

Rotation Angle: Allows rotation of Flare images~~.~~ Pick a degree.

Relative Distance (requires Size to be Relative): If Size is Relative, the Flare is treated as a 3D object~~.~~ If you specify that this distance is 10000, the program will create the Flare at the size a circle with a radius of 10000 units at the Flare's position in space would appear on screen~~.~~ This means that Flares will automatically decrease in size as they move farther from the camera~~.~~ This may not be accurate according to physics, but it's enormously useful~~.~~ Relative Distance, if active, still reacts to the Height and Width Percentage to find the Flare's final size.

X Offset: Allows you to offset the Flare from its normal position~~.~~. If 0.0, the Flare image is centered to the Flare's First Point horizontally~~.~~. If 1.0, the Flare image has its left edge at the Flare's First Point~~.~~. If -1.0, the Flare has its right edge at its First Point~~.~~. You may also use numbers larger than 1.0 and smaller than -1.0~~.~~. Rotation still occurs around the Flare's First Point, so you can rotate a Flare around its end points if desired.

Y Offset: Allows you to offset the Flare from its normal position~~.~~. If 0.0, the Flare image is centered to the Flare's First Point vertically~~.~~. If 1.0, the Flare image has its top edge at the Flare's first point~~.~~. If -1.0, the Flare has its bottom edge at the Flare's First Point~~.~~. You may also use numbers larger than 1.0 and smaller than -1.0~~.~~. Rotation still occurs around the Flare's First Point, so you can rotate a Flare around its end points.

Opaque (check box): Opaque requires Z Buffered to be ON - it's automatically turned on for you, but don't turn it off~~.~~. If you turn ON the Opaque gadget, the Flare will be treated as a transparent object that you can use to create overlapping Flares which partly or fully obscure other Flares and ~~polys~~polygons in the drawing~~.~~. It also means you can create Flares as dark objects if you want~~.~~. See additional discussion of Opaque Flares in the Flare tutorials.

Obscurable (check box): If ON, the Flare will be obscured by polygons that appear in front of it - depending on the transparency of the obscuring polygon(s)~~.~~. If OFF, the Flare will always be visible.

S Edges (check box): If ON, the Flare will completely disappear when the Flare leaves the visible render screen~~.~~. If OFF, the Flare image will remain on screen even if the Flare's First Point is not on screen.

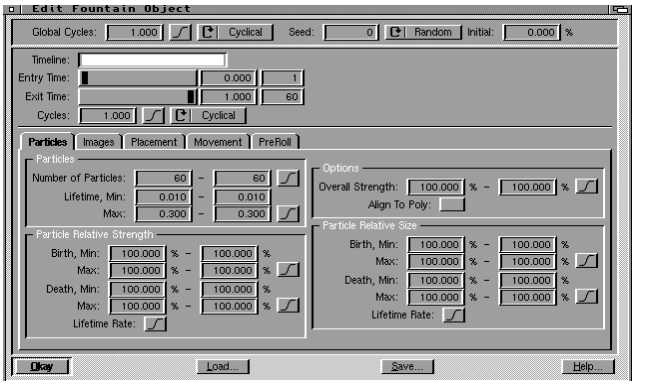
Z Buffered: If this is on, the Flare image is treated not as a Lens Flare source, but as a continuous source, so any occluding object can obscure only part of the Flare~~.~~. This allows you to place Flares behind objects, or even inside them~~.~~. See also Opaque Flares.

Obscured Trn: This only works if Obscurable is ON~~.~~. If ON (and if Obscurable is also ON), Flares react to transparent ~~polys~~polygons as though they were not there~~.~~. If OFF (and Obscurable is ON) Flares lose strength as they pass behind transparent ~~polys~~polygons~~.~~. The Flare reduction in strength is four times the normal reduction in strength from transparency~~.~~. A Flare disappearing behind a poly of transparency 192 or less completely fades~~.~~. A linear fade is accomplished between 192 and 255.

Size (~~rotary gadget~~cycle gadget): Relative or Fixed~~.~~. When Fixed, Relative Size above is inactive)

Limits: you cannot use this menu during facing or deform editing~~.~~. To be visible, Flares require a Texture List with an active bitmap selected.

Menu Item: Fountain (sub-menu: New or Edit)
Keyboard: (see sub-items)



Brief: Create or edit a Fountain Object.

Limits: you cannot use the Edit Fountain Object menu item during facing or deform editing. Fountains require a Texture List with an active bitmap selected.

This menu item allows you to either create a new Fountain or alter an existing one~~.~~.

Fountains require a Texture List to be assigned - so, after creating a Fountain, always assign it a Texture List. Also, the texture you assign must be a bitmap, See the tutorial on Fountains.

Fountains are shown in the Editor window as simple two-pointed polygons on screen. Point 0 of the poly is the source of the Fountain particles.

The New sub-item (Keyboard: <Shift> <Y>) creates a new Fountain at the current Attach Point. Edit (Keyboard: <Alt> <Y>) alters a previously existing Fountain. One Fountain must be selected before selecting Edit. Either way, this menu item opens the Edit Fountain Object requester. The gadgets in the window are:

Global Cycles: The number of times the Fountain's instructions will repeat during an animation.

Type (~~rotary gadget~~cycle gadget): Cyclical or Periodic. If Cyclic, the Fountain's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

Random Seed: This is the seed for the Fountain's pseudo-random number generators. Identical Fountains will generate identical particle distributions. Change this value for one or more of them to cause a different distribution, so they won't look perfectly choreographed - unlikely in nature. Also occasionally, you might have a Fountain that generates its particles in an "unfortunate" way. For instance, a single particle may suddenly flash directly in front of the camera, ruining an otherwise great scene. You can quickly alter this by changing the seed. Of course, this being a pseudo-random function, there's no guarantee that another particle won't flash directly in front of the camera - but usually this is a quick fix.

Seed Type (~~Rotary gadget~~Cycle gadget): There is no true randomness in computers. Particles are capable of using two types of pseudo-randomness in their behavior. If this gadget is set to Pattern, the randomness reveals a pattern underlying the random number generators common to the computer. This is most obvious when all of the particles travel the same distance. If Random, the particles use a factor based on their position in the Fountain queue to further randomize their behavior. It is still possible to detect a pattern now and then, but usually only during transitions in rotation values. Initial: A time offset (in percentage) for the Fountain's start. This is not a "PreRoll" offset. Instead, the Fountain will start at the value entered here (usually 0.0) and work up to the normal completion of the Fountain (always 1.0) then cycle back to the normal starting time of 0.0. Normal range for this variable is 0.0 to 1.0.

Time line: This is a graphic display box for the gadgets below it.

Entry Time: (slider) Sets the beginning time for the member. May be altered with the mouse.

Exit Time: (slider) Sets the ending time. May be altered with the mouse.

Cycles: The number of times the current member's parameters will be executed during the member's time allocation

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

Type: (~~rotary gadget~~cycle gadget) Cyclic, Periodic. If Cyclic, the member's instructions will execute from start to finish. If Periodic, the instructions will execute from start to finish and back to start.

Particles Tab~~(tab)~~

Number of Particles: The number of Particles that a Fountain sustains is set for each of Entry and Exit, for both Birth and Death maxima and minima, and can be relegated to CSpline control during animation by

clicking the Spline gadget near the text entry boxes~~.~~. If you're rendering only one picture, you do not need CSpline control.

Lifetime Min and Lifetime Max: The life of the particles specified as a percentage of the Fountain's allocated active time~~.~~. Normal values range from 0.01 to 2.0~~.~~. As a particle expires, it will be replaced by a new one to sustain the Particle Number specified~~.~~. A particle may be expired early if Particle Number decreases - say, during animation~~.~~. A random value between Max and Min will be used.

Particle Relative Strength (the next panel down) is also set for each of Entry and Exit, for both Birth and Death maxima and minima, and like many of the other controls in the Edit Fountain Object requester, can be consigned to the control of a CSpline by clicking the Spline gadget near the text entry boxes.

Lifetime Rate (Spline Gadget): Rate control for particle change between its assigned values

Options Panel~~Panel~~

Overall Strength: A multiplier for all particles' strength~~.~~. Normal value range is 0.0 to 1.0.

Overall Strength CSpline: Rate control for overall strength values.

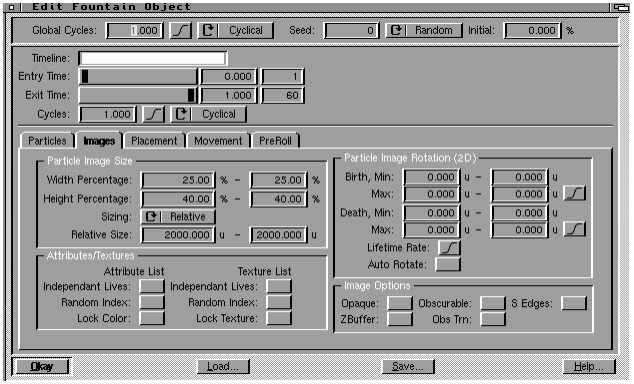
Align To Poly: If this is OFF, the fountain direction, rotation, etc are based on world coordinates. A direction of 0.0,0.0,0.0 is straight "up", or along the z axis in a negative direction. If this is ON, the fountain poly's second point is used to determine the up vector. This means you can tilt the fountain, or direct the particles visually by "aiming" the fountain in any direction. Note that the exact verticle direction is undefined, and if you put the second point so the vector is verticle, the default presentation is used.~~/// ??? 0501 /j~~

Particle Relative Size Panel~~Panel~~

Particle Relative Size (the bottom right panel) establishes sizes as a relative percentage of their original size for each of Entry and Exit, for both Birth and Death maxima and minima, and like many of the other controls in the Edit Fountain Object requester, can be consigned to the control of a CSpline by clicking the gadgets near the text entry boxes.

Lifetime Rate (Spline Gadget): Rate control for particle Relative Size

Images ~~(~~Tab~~)~~



Particle Image Size

Width Percentage: A multiplier for the width of the particle images on screen~~.~~. It is a percentage of the current render screen~~.~~. Used as aspect ratio control for particle images.

Height Percentage: A multiplier for the height of the particle images on screen~~.~~. It is a percentage of the current render screen~~.~~. Used as aspect ratio control for particle images.

Width and Height Percentages can be applied differently, depending on the state of the Sizing gadget~~.~~. See below.

Sizing: Relative or Fixed~~.~~. If Fixed, the Fountain's width/height percentage gadgets relate to Render Screen size~~.~~. If Relative, those percent gadgets relate to the Relative Distance gadget's value~~.~~. With Fountains, it is recommended to use Size as Relative.

Relative Size: Fountain size can be Relative or Fixed~~.~~. See the ~~rotary gadget~~cycle gadget to change this, if you need to~~.~~. If Fixed, the gadgets relate to the render screen~~.~~. If relative, the gadgets relate to the Relative Size gadget.

Attributes/Textures: This section provides check boxes for Attribute List and Texture List for each of Independent Lives and Random Index~~.~~. In addition, there is a gadget for Lock Color Attribute List, and one for Lock Texture under Texture List.

Particle Image Rotation (2D): The image rotation in degrees at particle birth~~.~~. A random value between Max and Min will be used, for each of Birth and Death.

Auto Rotate: If ON, the particle images will automatically be rotated so the "bottom" of the image always points to the source~~.~~. This is additive to the specified image rotation, so both can be used at the same time.

Image Options Panel~~:~~

Opaque: If this is ON, the fountain will behave as a transparent object~~.~~. Flares and Fountains are normally rendered as "additive" objects - their colors are added to the current color of the image at their location~~.~~. If you turn ON the Opaque gadget, the Fountain will be treated as a transparent object that you can use to create overlapping Flares and Fountains that partly or fully obscure other Flares, Fountains, and ~~polys~~polygons in the drawing~~.~~. It also means you can create Fountains as dark objects if you want~~.~~. See Transparent/Opaque Fountains in the tutorials section.

Obscurable: If ON, the particles will be obscured by polygons that appear in front of particles~~.~~. Note that this is dependent on the transparency of the obscuring polygon(s)~~.~~. If Off, the particles will always be visible.

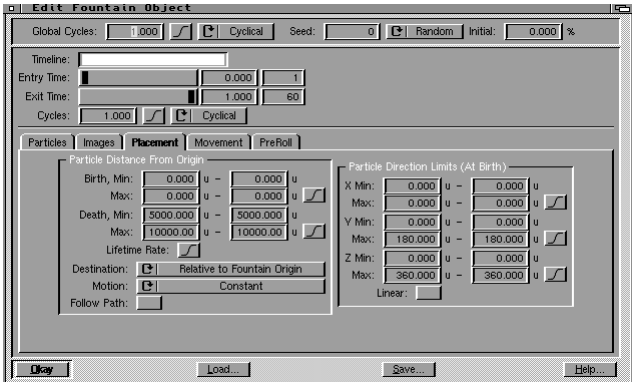
S Edges: If ON, the particles will completely disappear when the particle leaves the visible render screen~~.~~. If OFF, the particle image will remain on screen even if the particle is not on screen.

Zbuffered: If ON the particle images will be processed as part of the polygon database~~.~~. Allows partial obscuring of particle images.

Important: If you use this option, make sure the Fountain has a small number of particles, or prepare for a LONG render time.

Obscure Trn: (only works if Obscurable is ON): If ON (and Obscurable is ON), particles react to transparent ~~polys~~polygons as though they were not there~~.~~. If OFF (and Obscurable is ON) particles lose strength as they pass behind transparent ~~polys~~polygons~~.~~. The particle reduction in strength is four times the normal reduction in strength from transparency~~.~~. A particle disappearing behind a poly of transparency 192 or less completely fades~~.~~. A linear fade is accomplished between 192 and 255.

Placement ~~(~~Tab~~)~~



Particle Distance from Origin panel

Distance Birth Max/Min: The distance of the particle at its birth from the source in units in space~~.~~. A random value between Max and Min will be used.

Distance Death Max/Min: The distance of the particle at its death from the source in units in space~~.~~. A random value between Max and Min will be used.

This panel specifies distances measured for particles to begin and end their lives, measured from the origin~~.~~. The direction they travel is controlled by the ~~rotary gadget~~cycle gadget, which can be either Relative to Fountain Origin, or Relative to Birth Place~~.~~. It seems like a subtle difference, but it makes a big difference in the way a Fountain looks on screen.

Motion: This ~~rotary gadget~~cycle gadget can be Constant or Relative~~.~~. If relative, Distance values are treated as absolute positions in space, relative to the source~~.~~. If Constant, the original distance value is treated as an absolute position in space relative to the source, but the destination value is treated as an offset from the particle's initial position.

Also see the Motion option.

Follow Path: This boolean controls how the particles obey the fountain path assignments. The particles are created relative to the position the paths have moved the fountain to when the particle is born. If this is Off, then they are free of the path's instructions, remembering the position of the source at their birth time and rotating, drifting, etc. relative to that point, used for comets, fireworks, writing, etc. If On, the particles will continue to obey the path instructions, moving along relative to the fountain. They will rotate, drift, etc. relative to the changing source position. ~~/// ??? 0430~~

Particle Direction Limits panel

X Max/Min: The initial direction of the particles around the X axis~~.~~. A random value between Max and Min will be used.

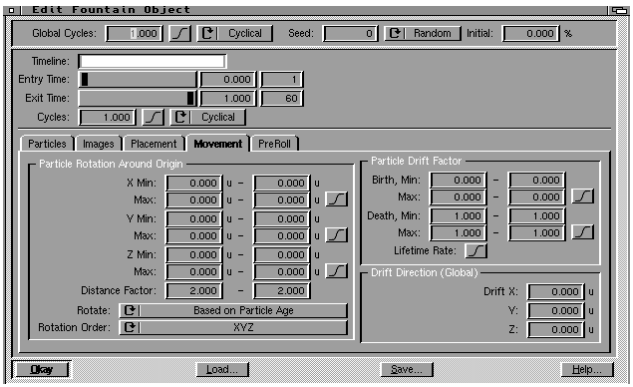
Y Max/Min: The initial direction of the particles around the Y axis~~.~~ A random value between Max and Min will be used.

Z Max/Min: The initial direction of the particles around the Z axis~~.~~ A random value between Max and Min will be used.

The CSpline objects place these parameters under non-linear CSpline control.

Linear: When you set your direction limits, as the fountain creates particles, they are given directions within these limits. Normally, (when this is OFF) this direction is a random value between the max/min limits in effect at the current time. If this is ON, the particles are given a direction that is a linear division of the angle between your max/min limits. For instance, if you had a range of 360 degrees and had 36 particles, they would be distributed at 0, 10, 20, and so on. This allows you to achieve perfect distribution around an arc, which is impossible if using randomness for the direction. ~~/// ??? 0430 /j~~

~~@ DISTANCE INDIVIDUAL CSpline: Rate control for particle change~~
~~@ between its assigned Orig/Dest values~~
~~/// Is this removed ??? 0430 /j~~



Movement ~~(tab)~~Tab:

Particle Rotation Around Origin panel

X Max/Min: An amount of rotation around the X axis that the particle will perform over its Life or as a function of its distance from the source (see Rotate Type gadget)~~.~~ A random value between Max and Min is used.

Y Max/Min: An amount of rotation around the Y axis that the particle will perform over its Life or as a function of its distance from the source (See Rotate Type gadget)~~.~~ A random value between Max and Min is used.
Z Max/Min: An amount of rotation around the Z axis that the particle will perform over its Life or as a function of its distance from the source~~.~~ See Rotate Type gadget)~~.~~ A random value between Max and Min is used.

Particle Distance Factor~~:~~

Rotate (Type): If Based on Age, particles will rotate based on their age, rotating from 0 to a random value between the limits you set in rotation Max/Min X, Y, and Z~~.~~ If Based on Distance, particles will rotate based on their distance from the source~~.~~ From 0 at maximum distance to a random value between the limits you set in rotation Max/Min when at the source.

Rotation Order: Allows you to specify an order for the rotations to take place, if you are using more than one axis of rotation in the Rotation X, Y, and Z gadgets.

~~@ ROTATION INDIVIDUAL CSpline: Rate control for particle change~~
~~@ between its assigned Orig/Dest values~~

Particle Drift Factor

CAUTION: Specifying very small particle values in both Max and Min will lengthen Fountain calculation times. Do not use values of 0.0 in both Max and Min.

A Fountain's Particle Drift Factor is established by two numeric entry boxesTwo numeric entry boxes establish a Fountain's Particle Drift Factor for each of Birth and Death minima and maxima, These gadgets specify multipliers for the global Drift values whichvalues that will be used by the particle at its birth. A random value between Max and Min will be used.

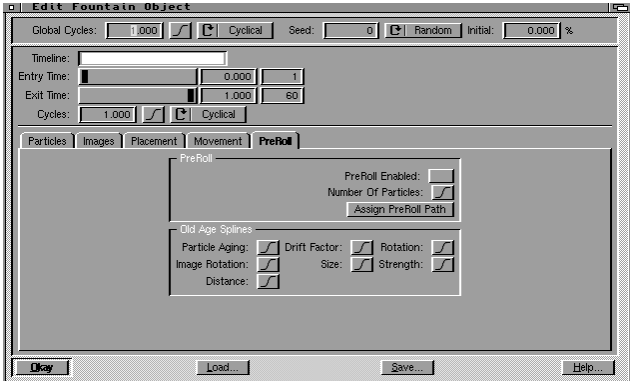
Drift Direction (Global)

Drift X: The global amount and direction of X drift that a Fountain applies to its particles.

Drift Y: The global amount and direction of Y drift that a Fountain applies to its particles.

Drift Z: The global amount and direction of Z drift that a Fountain applies to its particles.

PreRoll (Tab)



PreRoll exactly duplicates a Fountain's animation time period, but directly precedes it. It's all calculated before the Fountain begins rendering on Frame 1 of your animation. Use PreRoll to control the initial placement of particles at the beginning of an animation. In PreRoll, also, but you can have CSpline control to alter the particle behavior during the PreRoll time for the number of particles. The number of particles during the animation is controlled by the entries in the Edit Fountain Object requester. For the other CSplines in the PreRoll window, they will take effect only when a particle goes into "old age". This is when the particle becomes 1.0 or older, and can only happen when PreRoll is engaged.

For instance, if a Fountain has Entry/Exit number particle values of 1/10 and PreRoll is engaged, during PreRoll the particles are created. The first particle is created at 0.0, the second at 0.1, etc. By the time the animation starts, the first particle's age is 1.0, the second is 0.9, etc. So the first particle will immediately use the old age CSplines, but the second particle will not use them until it reaches 1.0. Notice that not only does PreRoll have to be ON, but the Life values have to be greater than 1.0 for the old age CSplines to take effect - otherwise the particles will expire before they get old enough to use them.

PreRoll Enabled: If ON, PreRoll is engaged; if OFF, it is not.

Number of Particles CSpline: Rate control for PreRoll Particle Number values. PreRoll uses the same Number of Particles values as the Fountain, but controls the particle number with this CSpline instead.

Assign PreRoll Path: During the PreRoll time, the Fountain can use a different path hierarchy than it uses during the animation. This gadget lets you pick a path for use by PreRoll. Click it, and the Edit Fountain Object goes away, giving you the opportunity to select a Path in your drawing for PreRoll to use. If you select no Path, or a poly that isn't a Path, the Edit Fountain Object returns, after an error message.

Old Age CSplines: This panel of the PreRoll tab determine how PreRoll is to handle aging during its operation.

Particle Aging: Rate control for particle aging when particles are 1.0 or older

Drift Factor: Rate control for Drift values when particles are 1.0 or older

Rotation: Rate control for Rotation values when particles are 1.0 or older

Image Rotation: Rate control for Image Rotation values when particles are 1.0 or older

Size: Rate control for Size values when particles are 1.0 or older

Strength: Rate control for Strength values when particles are 1.0 or older

Distance: Rate control for Distance values when particles are 1.0 or older

~~Limits: you cannot use the Edit Fountain Object menu item during facing or deform editing. Fountains require a Texture List with an active bitmap selected.~~

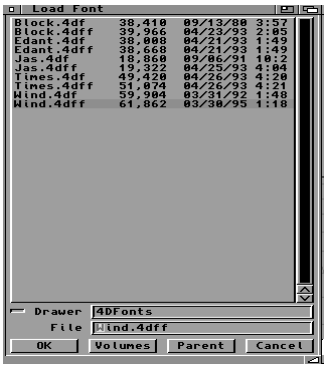
Menu Item: Extern (Handlers)
Keyboard: None

Brief: Provide menu access to external handlers for special functions

The Object / Extern / New menu sub-item opens the Choose Handler requester for External handlers~~.~~ Possibilities include: Brickwall, drawnote, etc~~.~~ These are contained on the distribution disk set and copied to your hard drive during the install procedure~~.~~ You can add more ~~- downloaded from the Internet,~~ supplied by Nova Design or third parties~~, or perhaps obtained from computer BBS's.~~ New opens the requester to load one of these~~.~~ Edit lets you establish the parameters that relate to an Extern handler that already exists.

Extern Handlers, or "hooks" are ways for external programs to alter an Aladdin 4D database - that is, a drawing~~.~~ These are similar to External Tools, but Extern Handlers are able to do far more~~.~~ The main difference is that the Extern Handlers can alter the database BETWEEN frames of an animation~~!~~! This presents many new abilities in rendering.

The following Extern Handlers are included on the distribution disks:




```
/// FLAG: Brickbuild handler /j
/// handler not finished yet 0505 /j
@ BRICKBUILD HANDLER:
@
@ This example handler nicely introduces the purpose of Extern Handlers in
@ Aladdin 4D. It builds brick walls over time, offering
@ many user definable options, plus the ability to use
@ textures, Attribute Lists, etc.
@
@ To use the BrickBuild handler, choose the Object menu item
@ Extern / New. When the requester opens, use its About button to read
@ the documentation for more instructions.
@
@ Handlers are designed to work in this way. Documentation for using each
@ is included with the handler, accessible from its About button.
```

DRAWNOTE HANDLER:

This Extern Handler is a way to include any number of variable-length text files in a drawing. This means you can create notes directly in your drawings, to remind yourself about what you did, or what you meant to do, settings for certain attributes, etc. These notes are saved and loaded with the drawing. You get to use your choice of editors and readers for the notes.

Menu Item: Text (sub-menu)
Keyboard: (see sub-items)

Brief: Write text as Aladdin 4D objects, load, save and modify fonts

This menu item gives you access to Aladdin 4D fonts. Its sub-items are:

Open Font: Opens the Load requester for an Aladdin 4D vector font Aladdin 4D uses two basic types of fonts, distinguished by the filename extensions ".4df" and ".4dff." Normally you'll see two fonts by the same name with the two different extensions in the fonts directory. The difference between these fonts is that ".4df" is meant for desktop publishing, such as output to Professional Draw clips, using compound objects, and ".4dff" is meant for video as rendered by Aladdin 4D. Compound objects account for holes in the characters - like "O" in a special way in Professional Draw. In general, always use the .4dff format, unless you intend to output Professional Draw clips.

Save Font As: Opens the filename Save requester to save a font. You can define letters (see "Define letter" to create the font information to be saved. This sub-item will open the file requester allowing you to save the current font, either by the same name, or by a new name. Note the naming convention in Load Font - not mandatory, but suggested.

Flush Font: Removes a font from memory (not disk). This sub-item will clear the currently loaded font from memory, freeing that memory for use by the system.

Define Letter (keyboard: <Alt> <T>): Pick an object or polygon, and make it a letter in the current font. The object doesn't have to LOOK like a letter, even. In this way you can use Aladdin 4D's font structure

to create easily accessible basic shapes from which you can build strings just by typing.. Define Letter replaces a letter in the currently loaded font.. Select the polygon(s) that you wish to use for a letter and then select this sub-item.. The Define Letter window opens.

Simply type the letter you wish the polyspolygons to represent and OK the requester.. If no font is currently loaded when you choose this sub-item, you will get the New Font Name requester.. Enter a name for the font and OK the requester.. The new font will be created, and the Define Letter requester will open.. If the letter you enter in the Define Letter requester is already defined for the font, you'll be asked whether to overwrite the letter's definition.. If you choose to overwrite, the old letter will be replaced in the font with the new polygons.. After defining one or more letters, you can save the font to disk under any file name you choose.. It does not have to be the same name you have given to the font, nor the one by which it was loaded.. Special polyspolygons may not be used for font letter definitions.

Write (Keyboard: <Shift> <T>): This is where you enter the text to be created as an object.. A font must be loaded before you use this item.. The option brings up the Write Text requester.

This window's gadgets are:

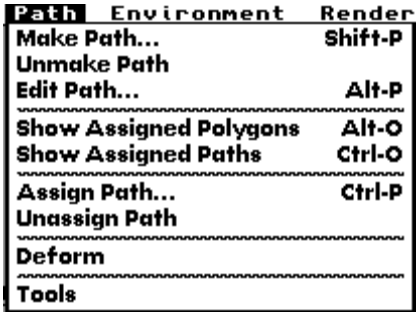
Current Font (informational only, not editable): displays the name of the currently loaded font.. Spacing: The distance to be placed between each letter

Position: The Z coordinate that will act as the baseline for the letters

Text: (string entry): Enter here the text you wish to make into objects.

Limits: You may not choose this menu item when in deform or facing mode.. Font format used does not record Texture List assignments, Attribute List assignments, or shading information.. Only 128 characters are used in the font format.. Special polyspolygons may not be used for font letter definitions.

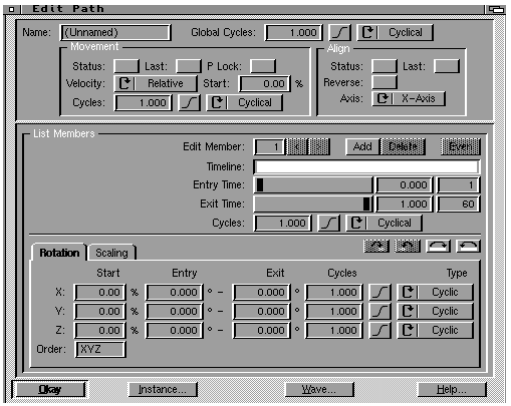
The Path Menu



Menu Item: Make Path
Keyboard: <Shift> <P>

Brief: Convert selected polygon to a path

This menu item allows you to make the selected polygon into a path~~.~~. To use it, you must have one (and only one) polygon selected~~.~~. Selecting the menu item opens the Edit Path window.



The gadgets in this window are:

Name: You can give paths names, the better to keep them sorted out and logically distinguishable from each other.

Global Cycles: The number of times the Path's instructions will be executed.

Spline Gadget: Turns the Path's activities over to the control of a CSpline~~.~~. The members list does the same thing as the CSpline, but in a linear manner.

Type: Cyclical, Periodic (~~rotary gadget~~cycle gadget)~~.~~. If Cyclical, the list will execute from start to finish~~.~~. If Periodic, the list will execute from start to finish and back to start.

Movement Panel

The Movement gadgets control translation (movement) along the path.

Status: If this is ON, the Path's movement is enabled~~.~~. This status on/off switch lets you "turn off" a path's behavior if you need to, without having to go back through the whole settings process to turn it back on~~.~~. If Status is OFF, no translation occurs, just rotation (if specified).

Last: If ON, the "last" side of the polygon representing the path will participate in the translation~~.~~. If OFF, it will not.

Movement Type: If Cyclical, the translation occurs from first point to last point~~.~~. If Periodic, the translation occurs from first point to last point and back to first point.

P LOCK (Point Lock): If ON, translation is from point to point only~~.~~. No translation occurs between points~~.~~. If OFF, the translation occurs smoothly along all sides of the polygon representing the path.

Velocity: If Relative, each segment of the path is given an equal amount of time~~.~~. Shorter segments will move the polygons slower, and longer segments will move them faster~~.~~. If Constant, the length of time is divided by the total length of the path and each segment is given a proportional share~~.~~. Shorter and longer segments will move the polygons at the same speed.

Cycles: The number of times the translation (movement) occurs

Start %: If a value is entered here, the translation will start as though the animation has completed this amount of time and cycle back to this position. For example, if you specify 0.25 here, on the first frame of the animation the translation will already be 25 percent along the path, and will continue until it comes back to this position by the end of the animation.

CSpline Gadget: Allows you to choose a CSpline for controlling translation

Cyclical, Periodic (rotary gadgetcycle gadget): If Cyclical, the translation will execute from start to finish. If Periodic, it will execute from start to finish and back to start.

Align Panel

The Align section of the requester allows you to specify that the polygons assigned to the path be automatically rotated by the direction of the path. Polygons are rotated from a fixed angle. You may want to rotate the polyspolygons by an intermediate path, to get the desired look.

Align / Status: If ON, the polygons assigned to the path will be automatically rotated to match the angle of the segment of the polygon representing the path that is active. If OFF, they will not be automatically rotated.

Reverse: If ON, the rotation will be opposite that dictated by the direction of the segment. If OFF, it will be that dictated.

Last: If ON, the Last Segment will participate in the rotation; if OFF, it will not.

Axis: The axis that the direction will be measured against. Normally, this is the Z axis.

List Members

Aladdin 4D's "Member" section in its requesters permits assigning time slices of an animation to different specifications, even assigning them unequal portions of the animation. This control can also be turned over to a CSpline for smoothly transitioned time slicing of the application of various parameters. The "List Members" area requester does the same thing as the CSpline, but in a linear manner. Initially, the requester contains only one member, which is assigned the entire time line (see the graphic Time line gadget in the requester). You can add members, name them, and cycle through them to observe their whereabouts on the Time line. Each can have different specifications in the requester, and as you step through them, these specs will appear in the requester's gadgets.

Edit Member: changes the member you're working on.

Add: Makes a new member

Delete: Removes rid of the currently selected member

Even: Causes an even distribution of time between all members

Entry Time: (slider) Sets the beginning time for the member. May be altered with the mouse.

Exit Time: (slider) Sets the ending time. May be altered with the mouse.

Cycles: The number of times the current member's parameters will be executed during the member's time allocation

Spline Gadget: Invokes the CSpline editor and turns control over to it from the linear members.

Member type: (~~rotary gadget~~cycle gadget) Cyclical, Periodic~~.~~. If Cyclic, the member's instructions will execute from start to finish~~.~~. If Periodic, the instructions will execute from start to finish and back to start.

Rotation Tab~~(tab)~~

This panel contains rotation instructions and their modifiers for the currently selected member~~.~~. Each axis has four associated numeric entry gadgets, one each for rotation instruction at Start, Entry, Exit, and Cycles, in addition to a CSpline gadget and a ~~rotary gadget~~cycle gadget where you can choose Cyclic or Periodic.

Start %: In the beginning...~~.~~. If entry is 0 and exit is 360, a start percent of 0 will rotate from 0 to 360 degrees~~.~~. A start percent of 0.25 will rotate from 90 through 360 to 90 degrees~~.~~. Use it as an offset in global time~~.~~. An example would be to set up one leg of an animal, then clone the leg and alter all paths in the cloned leg to a Start percent of 0.5.

Entry: The rotation of the assigned ~~polys~~polygons at the entry time of the member

Exit: The rotation of the assigned ~~polys~~polygons at the end of the member

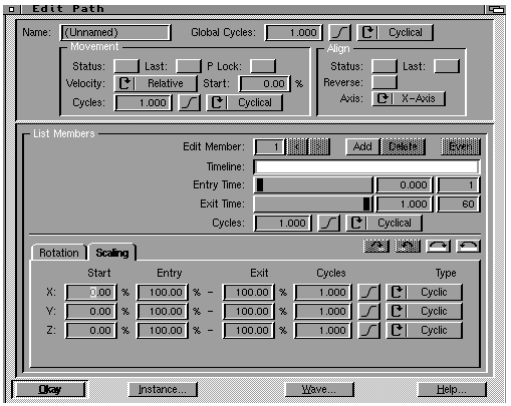
Cycles: The number of times the difference between entry and exit rotations will be performed.

CSpline: Allows use of a CSpline for rotation instructions in each axis

Type: If Cyclic, the rotation will occur from entry to exit~~.~~. If Periodic, the rotation will occur from entry to exit and back to entry.

Order: Specify the order rotations will occur~~.~~. The possibilities are XYZ, XZY, YXZ, YZX, ZXY, ZYX.

Scaling Tab~~(tab)~~



Paths can be configures to perform scaling "on the fly"~~.~~. This tab controls that feature~~.~~. The gadgets in it are pretty much the same as those for Rotation, but in this tab, they refer to changing the size of the objects that are assigned to the path, instead of rotation~~.~~. You can, of course, have both scaling and rotation.

Each axis has four associated numeric entry gadgets, one each for Scaling instruction at Start, Entry, Exit, and Cycles, in addition to a CSpline gadget and a ~~rotary gadget~~cycle gadget where you can choose Cyclic or Periodic.

Start %: If entry is 0 and exit is 360, a start percent of 0 will rotate from 0 to 360 degrees~~.~~. A start percent of 0.25 will Scale from 90 through 360 to 90 degrees~~.~~. Use it as an offset in global time.

Entry: The Scaling of the assigned ~~polys~~polygons at the entry time of the member

Exit: Scaling of the assigned ~~polys~~polygons at the end of the member

Cycles: The number of times the difference between entry and exit Scaling specifications will be performed.

CSpline: Allows use of a CSpline for Scaling instructions in each axis

Type: If Cyclic, the Scaling will occur from entry to exit~~.~~. If Periodic, the Scaling will occur from entry to exit and back to entry.

Instance ~~(button)~~Button

Instances are work savers~~.~~. They are also memory eaters, but that's another story~~.~~. An "instance" of a path is a total, complete copy of a path~~.~~. When Aladdin 4D gets down to rendering your picture or animation, it consults the path's instructions for Instances, and if it finds any, it adds them to the list of things in its database to be rendered~~.~~. So, you can basically make one set of objects, and instance them as many times as your Amiga has memory to accommodate, to get the "cast of thousands" you desire~~.~~. Saves work over making thousands of objects, no?

In the Edit Path requester, whether you've got to it with Make Path (a new one) or Edit Path (editing an existing one) Instances are accessed with a button in the bottom border.

When you're done setting up the Path as you want it, click the Instance button to multiply it as needed~~.~~. The action brings up the Edit Path Instances requester~~.~~. Here's what it contains:

Number of Instances: This is the number of instances of the assigned ~~polys~~polygons (and paths) that will be created~~.~~. If you use 2, you will have two instances and the original object for a total of three objects.

CSpline gadget: Allows you to use a CSpline to control all entry and exit values at once in a non-linear manner.

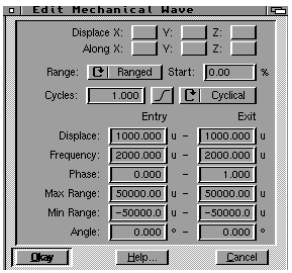
The requester also provides Entry and Exit numeric entry boxes for each of Offset Time, Offset axes (X, Y, and Z), and Rotate axes.

Offset Time: If you have paths being instanced, this allows you to equally apply offsets in time to the instances~~.~~ ~~.~~ For example, if you have a path rotating an object from 0 to 360 degrees, and you enter a time offset of 1.0, with 3 instances, on the first frame of the animation the original object will be rotated 0 degrees, the first instance will be rotated 90
degrees, the second instance 180, and the third instance 270 degrees~~.~~ ~~.~~ Similar effects are achieved in translation, scaling, etc~~.~~ ~~.~~ An optional CSpline allows you to control the time changes non-linearly.

Offset X, Y, and Z: The distance in which the instances will be equally spaced along these axes~~.~~ ~~.~~ Each has an optional CSpline to allow you to alter the time the change in offset between entry and exit values occurs.

Rotate X, Y, and Z: The angle in which the instances will be equally spaced along these axes~~.~~ ~~.~~ Each has an optional CSpline to allow you to alter the time the change in rotation between entry and exit values occurs.

Waves ~~(button)~~ Button



In the Edit Path requester, whether you've got to it with Make Path (a new one) or Edit Path (editing an existing one) you can set up Waves for your Path by clicking the Wave button in the bottom border~~.~~ ~~.~~ These controls are for mechanical waves generated in the path directly assigned to the path being altered only~~.~~ ~~.~~ The polygons will have their actual point positions changed~~.~~ ~~.~~ They should, in general, be triangles, since use of Waves will almost always change them so they are not flat and therefore would not render properly~~.~~ ~~.~~ Also, when using Waves along only one axis, rectangular meshes give best results~~.~~ ~~.~~ When using Waves along two or three axes, concentric rings of polygons with the Wave path's First Point at the center of the rings give best results.

This button lets you set up Waves~~.~~ ~~.~~ It opens the Edit Mechanical Wave requester~~.~~ ~~.~~ The gadgets in this window are:

Displace X, Y, and Z: Select one or more axes to use to displace polygons.

Along X, Y, and Z: Select one or more axes to use with range gadgets~~.~~ ~~.~~ This selection works hand in hand with the Displace selection~~.~~ ~~.~~ For example, you may choose to Displace Z along X, which would create a wave along the X (parallel to Y) that displaced the points of the ~~polys~~polygons in the Z direction~~.~~ ~~.~~ If you choose to Displace Z along the X and Y, the wave will be circular in the XY plane, and the points will be displaced in the Z direction (like a cylinder with its long axis on Z)~~.~~ ~~.~~ Different combinations of these gadgets will give all possible longitudinal, circular, and spherical waves.

Important: If no Displace XYZ and Along XYZ are entered, no wave will be generated.

Range (~~rotary gadget~~cycle gadget): Clicking this gadget will select either Ranged or Infinite~~.~~ ~~.~~ If Ranged, the values in the range gadgets will be used~~.~~ ~~.~~ A drop-off of displacement according to the square of the distance is applied~~.~~ ~~.~~ If Infinite, the values in the Range gadgets will be ignored, and the waves will be distributed uniformly in space.

Start %: An optional offset in time for Waves, similar to Start % on Translation and Rotation

Cycles: The number of times the difference in Entry/Exit values will occur.

Type (~~rotary gadget~~cycle gadget: Cyclical, Periodic): If Cyclic, the Wave's instructions will execute from start to finish~~.~~ If Periodic, the instructions will execute from start to finish and back to start.

Spline Gadget: Invokes the CSpline editor and turns control over to it

Entry and Exit Settings for each of:

Displace: The amount to displace a polygons' points.

Frequency: The "closeness" of the waves.

Phase: Waves are basically sinusoidal functions, with low and high peaks that can move over time~~.~~ Phase will cause the wave to move~~.~~ Normal operation uses an Entry Phase of 0.0 and an Exit Phase of 1.0~~.~~ This will cause the waves to move the distance on the polygon determined by their frequency toward the center of the wave~~.~~ Values of 1.0 and 0.0 cause the wave to move outward from the center.

Phase refers to the position of the crest~~.~~ Use 0.0 to 1.0 to move the waves from crest to crest~~.~~ Use 0.0 to -1.0 to reverse their direction~~.~~ Use 0.0 to 3.0 to make them move three times as fast.

Max Range: The largest distance that will have any displacement if Range type is set to Ranged.

Min Range: The smallest distance that will have any displacement if Range type is set to Ranged~~.~~ This is normally set to the negative value of Max Range~~.~~ If you use a circular wave and set Max Range to 50000 and Min Range to -50000, the wave will have its strongest displacement at the center of the circle and gradually weaken as it reaches 50000~~.~~ If you set Max Range to 50000 and Min Range to 0, the wave will have its strongest displacement at 25000 and weaken in both directions (toward 50000 and toward 0) creating a ring of displacement~~.~~ If you set Max Range to 50000 and Min Range to 40000, the wave will have its strongest displacement at 45000 and weaken to 0 at 50000 and at 40000, creating a thinner ring~~.~~ By using proper Entry/Exit values, you can make a ring expand during the animation~~.~~ Remember circular waves should be used on circular rings of ~~polys~~polygons~~.~~ Longitudinal waves can be similarly controlled, allowing you to create long bands of wave influence.

Angle: An optional angle to rotate the influence of longitudinal and circular waves.

Limits: You cannot use the Path / Make Path menu item during facing or deform editing~~.~~ Special ~~polys~~polygons, such as camera, target, gas, light, and waves cannot be turned into paths~~.~~ Splines cannot be paths (convert them into multi-sided ~~polys~~polygons first, then store the splines in a drawing Space)~~.~~ If you jump paths, which are linked to other paths, to another space, the links must be broken~~.~~ You will be asked for confirmation before this occurs.

Menu Item: Unmake Path
Keyboard: None

Brief: Convert path to regular polygon

This menu item allows you to get rid of a selected path and turn it back into a regular polygon~~.~~ To use, you must have a single path selected~~.~~ Any other paths or polygons that are assigned to the path will be, of course, unassigned.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Alter Path
Keyboard: <Alt> <P>

Brief: Alter an existing path

This menu item allows you to edit the instructions of a path~~.~~. To use, you must have a single path selected~~.~~. When you choose the menu item, the Path window opens~~.~~. See Make Path for a full description of this window and its gadgets.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Show Assigned ~~Polys~~Polygons
Keyboard: <Alt> <O>

Brief: Show polygons that are assigned to a specific path

This menu item allows you to see which polygons are assigned to a specific path~~.~~. To use, select the path, then choose the menu item~~.~~. ~~Polys~~Polygons that are assigned to the path will select.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Show Assigned Paths
Keyboard: <Ctrl> <O>

Brief: Show paths that are assigned to a specific path

This menu item allows you to see which paths are assigned to a specific polygon~~.~~. To use, select the path, then choose the menu item~~.~~. ~~Polys~~Polygons that are assigned to the path will select.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Assign Path
Keyboard: <Ctrl> <P>

Brief: Assign selected objects to a path

This menu item allows you tell the program which polygons are to be assigned to which path. To use, just select the polygons (or camera, target, Gas, Light, Wave, etc.), then select the menu item. After you're requested to select the path you want them assigned to, select a path by one of its points.

If the polygon or object had a previous path assignment, it will be changed to the new choice.

An object cannot be assigned to two paths at once. If you want multiple layers of motion, assign the object(s) to a single path, then assign that path to a second path. This gives you control over multiple motions for any object (s).

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Unassign Path
Keyboard: None

Brief: Break path assignments

This menu item allows you to tell polygons or objects not to look to a path for instructions. Just select the polygons or objects and select this menu item. They will no longer have path assignments. If they had deform levels, these will also be removed from memory and cannot be retrieved.

Limits: You cannot use this menu item during facing or deform editing.

Menu Item: Deform (sub-menu)
Keyboard: (with sub-items)

Brief: To establish and edit deforms (morph / keyframe animation)

This menu allows you to control keyframe type deforms of polygons assigned to a path. It has these sub-items:

Begin Level (<Shift> <K>)
End Level (<Ctrl> <K>)
Edit Levels (<Alt> <K>)
First Level Position
Delete All Levels

Here is what they do:

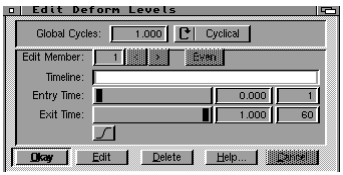
Begin Level: The Deform option in Aladdin 4D implements "keyframe" morph-like animation. To operate, it needs to know where the objects start, and where they end. Deform levels are graphic, for the most part. You get to move the objects around between keyframe definitions, to tell the program what you want. Begin Level starts a Deform definition. This informs the program that you want to add a level of Deform to a path's assigned polygons. First select the path. The path must have polygons already assigned to it. Then select the Begin Level menu item. When you select it, the program records the position of all points in the polygons assigned to the selected path. Move the polygons to the new desired position. You can move a single point, poly, all ~~polys~~polygons, resize, rotate, etc. When satisfied,

select the End Level sub-item~~.~~. The polygons will return to their starting positions, but all points will be recorded as the "destination" deform level.

Keyframe deforms are linear only~~.~~. If you rotate a sphere, for example, 180 degrees, and use this as the new position, the sphere will not rotate during the animation, but instead will turn inside out~~!~~! If you want it to rotate during an animation use the path's rotation instructions~~.~~. Deforms may, however, be used for translation, but they require more memory than normal translation.

End Level: You can have many key frames - that is, Deform levels~~.~~. Each time you select this item, a new level will be added to the deform instructions~~.~~. You will see the polygons take the position of the last level when you choose this menu item - if there are other deform levels~~.~~. You do not have to select the path since you are adding a level and the program knows which path you are adding the level to~~.~~. You must be adding a level to use this menu item.

Edit Level (sub-menu): To edit the Deform levels of a path select it, then choose this menu item~~.~~. The Edit Deform Levels window opens.



Its gadgets are:

Global Cycles: The number of times the level will be executed.

Cyclical, Periodic (~~rotary gadget~~cycle gadget): If Cyclic, the level will be executed from start to finish~~.~~. If Periodic, the level will be executed from start to finish to start.

Edit Member: This is the number of the Deform level you want to edit~~.~~. You can type it or change it with the clickable gadgets~~.~~. Also, the <Up> <Down> cursor keys may be used to change levels.

Entry Time (slider): A visual representation of the currently selected level's beginning time~~.~~. May be edited with the mouse (LMB).

Exit Time (slider): A visual representation of the current level's ending time~~.~~. May be edited with the mouse (LMB).

EVEN: Distributes time evenly between all levels

Delete: Delete the currently selected level

Spline Gadget: Turns Deform control over to a CSpline.

When you close the requester, returning to the Editor, the polygons of the path change to the level's position~~.~~. The program enters level editing mode~~.~~. You may change the polygon's positions for this level~~.~~. When finished, you must select "End Level".

First Level Position: This menu sub-item returns the Editor to the positions of the first Deform level.

Delete All Levels: Does just that~~.~~. Gets rid of all Deform definitions.

Limits: You cannot use this menu item during facing~~.~~. Special ~~polys~~polygons that are not visible in the render do not acquire deform levels.

Tools (sub-menu)

The Path / Tools menu has two sub-items: Align to Path, and Path Extrude~~.~~. These are special, path-related tools, hence their individual treatment.

Menu Item: Tools / Align to Path
Keyboard: None

Brief: Align selected ~~polys~~polygons along a path.

Although it may be put to general use for your own creative ideas, this tool is designed specifically to align extruded fonts along a path~~.~~. It does not operate on splines~~.~~. The Path to be aligned must be a multi-sided poly. (see "SPOTOPOL" to convert splines to ~~polys~~polygons, if you need to)~~.~~. ~~Polys~~Polygons must also be grouped in separate "objects".

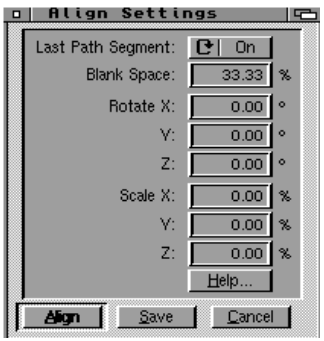
The tool is designed to think of positive Z as "UP"~~.~~. You should work with your extruded letters facing you from a flat Y view~~.~~. The path should be drawn from the flat Z view~~.~~. The path does not have to be flat - the letters will align with vertical deviations.

The align will center to a point half way along the length of the path as described under the Blank Space option below.

If you want an arc that reads from the front, use this same orientation, but use a Rotate X of 90 (or neg 90), then use the normal Rotate function to Rotate the arc into position.

If you perform the align, and the text reads the opposite way of that you desire, use <Esc> to undo it before issuing the Set command, then use the Edit menu item "Reverse Points" to swap the point order along the path~~.~~. When you perform the align again, it will read the way you want.

The spacing between the object groups along the Align-To Path is relative to that along the X axis of the groups before the align~~.~~. Vertical spacing is kept constant only for the individual groups, so if you have several lines of text, you must align them in separate operations, or they will converge vertically.



Align to Path opens the Align Settings requester, which has these gadgets:

Last Path Segment (~~rotary gadget~~cycle gadget): If ON, the last segment of the path poly will participate in the align~~.~~. If OFF, it will not.

Blank Space: When you select the path poly to align to, the program will measure its length, then subtract the percentage of this length you specify here~~.~~. The ~~polys~~polygons will then be resized to fit the remainder of the path length, and moved into position along this remainder.

The polygons are ALWAYS CENTERED to the center of the path as measured from the path's First Point to its Last Point if Last Path Segment is OFF, or back to the First Point if Last Path Segment is ON.

Rotate X,Y,Z: Specify any angles of rotation you want~~.~~. This function is designed to work with the positive Z axis "up"~~.~~. This makes it easy to generate logos like those in the Universal Pictures ~~//XRef trademark /j~~ movie header~~.~~. You can also generate arcs that go overhead and read properly from the front by specifying an X rotation of 90 , or negative 90 depending on the side you want to view it from.

To achieve these effects, after performing the align, rotate the aligned letters as a unit around the X 90 (or -90) using the standard Rotate tool~~.~~. Partial rotations are also quite useful for special positioning.

Scale X,Y,Z: Specify any scaling values you want~~.~~. Unity is 1.0 - no change~~.~~. 2.0 would be twice, and 0.5 would be half the original size~~.~~. The objects will automatically be scaled to fit the length of the path after the Blank Space has been subtracted~~.~~. You may, however, want to reduce them even more, or enlarge them, in any of the three axes~~.~~. This can adjust the "letter spacing", or cause them to fit the smallest curve in the path better, or even cause them to overlap for some interesting effects.

Preparing ~~Polys~~Polygons for Align To Path Operations

As you know, "objects" in Aladdin 4D are composed of single polygons~~.~~. If the Align tool were to Align single polygons, the extruded letters would be reduced to random single ~~polys~~polygons along the alignment path~~.~~. You need to let the program know what you consider to be single objects~~.~~. Do this by grouping.

When the tool runs, it resizes ALL THE SELECTED ~~POLYS~~POLYGONS IN EACH GROUP it finds around the group's center, and rotates and translates the group by its center~~.~~. The tool looks in the currently active group level~~.~~. This means you do not have to have the extruded letters align, but can also align complete words, or even phrases if desired.

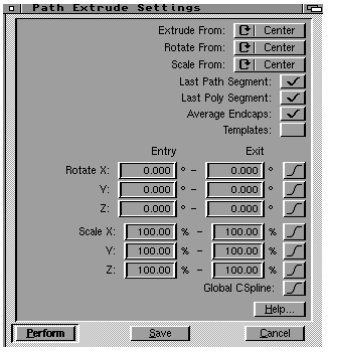
To group the letters into separate groups, you must select all the ~~polys~~polygons that compose a single letter, and group them.

To help you accomplish this, the Extrude tool has an option called Single Groups~~.~~. If on, each selected poly, when extruded will be given a separate group, along with its extruded shell and cloned face~~.~~. This is done only in GROUP LEVEL 3~~.~~. You may still have to select and group a few, for instance, the "stem" and the "dot" of the letter i.

Menu Item: Path Extrude
Keyboard: None

Brief: Create complex objects by extruding ~~polys~~polygons along a path

The Path Extrude tool lets you "lathe" an object along a path, making a solid object as it goes~~.~~. For many purposes, this operation might be much quicker and less complicated than obtaining the same shapes by using several invocations of other tools and options~~.~~. To use, create a path in the shape you want~~.~~. For example, you can use the outline of a letter~~.~~. You must also make a polygon that is the cross-section of the resulting object.



When selected, the option opens the Path Extrude Settings requester.

Its gadgets are:

Extrude From: (~~rotary gadget~~cycle gadget) Center, ATP - select the object's geometric center, or the current Attach Point.

Rotate From: (~~rotary gadget~~cycle gadget) Center, ATP~~.~~. As the poly follows the path, it rotates according to this specification - its center or the current Attach Point.

Scale From: (~~rotary gadget~~cycle gadget) Center, ATP - scaling on the "Fly" is from either geometric center or the current Attach Point.

@ Last Path Segment (check box): If On, the last segment of the path poly will generate a segment in the extrude. If Off, the last segment of the path poly will not generate a segment in the extrude.
@
@ Last Poly Segment (check box): If On, the last segment in the polygons being extruded will be extruded. If Off, they will not, resulting in an opening along the length of the extrude.
@
@ Average Endcaps (check box): This only applies if the last path segment is off. You can then specify whether you want the endcaps to be angled to fit the path, or set perpendicular to the last (and first) segments of the path.
@
@ Templates (Check box): If Off, the extrude is a hollow shell. If On, the extrude is a series of replicated polygons, like the fuselage formers in an aircraft fuselage.

Spline Gadget: Turns the option's controls over to a CSpline for non-linear control over time in an animation.

@ Rotate X, Y, and Z: Specify any angles of rotation you want.
@
@ Scale X, Y, and Z: Specify any scaling values you want. Note that 100% is unity, or no change. 200% would be twice as big, and 50% would be half the original size.
@
@ Global CSpline (SPLINE GADGET)

/// Need info on this option... not in externals help files ??? /j

The Environment Menu

Menu Item: Global Lights
Keyboard: <L>

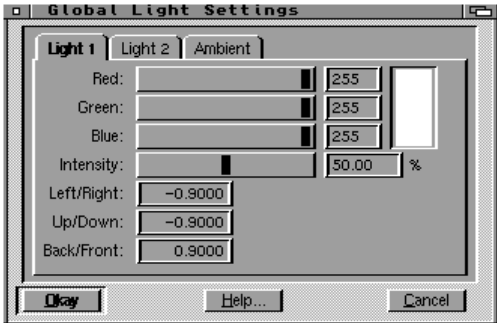
Brief: Control lighting globally

This menu item provides access to the global Light controls. Global Lights are slightly faster than local lights, but they cast no shadows.

Global Lights are quite different from the local lights defined in the Editor. They are directional only. Having no position in space, they throw a uniform amount of light onto your polygons no matter where they are. You can render using only Global Lights, or use them as in combination with local lights. Of course, you can also set Global Lights' intensity to 0.0 and thus use only local lights in your rendering.

There are two Global Lights, plus an Ambient Light setting. These are called Light 1, Light 2, and Ambient in the requester.

Think of the Global Lights as being attached to your camera. Everywhere the camera moves, the light keeps the same relative direction to your line of sight. This is true even when not using a camera, rendering the Editor's view.



When you select the item, the Global Light Settings requester opens.

Light 1 Tab~~(tab)~~

Color: This area has sliders (0-255) for choosing the component colors of the light: 0 to 255 levels each of Red, Green, Blue

Intensity (0-100%)

Left/Right: The Left/Right gadget controls intensity in the left to right direction,

Up/Down: Intensity in the above to below direction

Back/Front: Intensity controls in the front to back direction.

Light 2 Tab~~(tab)~~

Color: This area has sliders (0-255) for choosing the component colors of the light: 0 to 255 levels each of Red, Green, Blue

Intensity (0-100%)

Left/Right: The Left/Right gadget controls intensity in the left to right direction,

Up/Down: Intensity in the above to below direction

Back/Front: Intensity controls in the front to back direction.

Ambient Tab(tab)

The intensity of Light 1 and Light 2 may range from 0.0 to 1.0.

Ambient light is the amount of light that just "exists." It has no direction and no location in space, and it's generally quite diffuse. There are two uses for this light. First, it is used as fill, to keep darker areas from going completely black. Second, it's used when you want a completely bright image, by setting its intensity to 1.0.

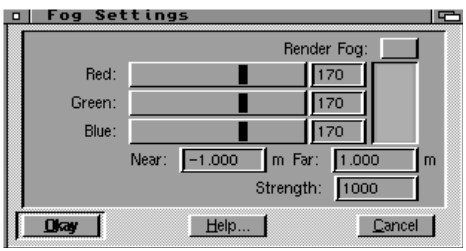
The more Ambient light a scene has the higher "key" the image will have. A dark, moody image would have little Ambient, but a bright, happy place might have substantial amounts of it. High levels of Ambient light generally mean lower image contrast, and low Ambient light generally produces a picture with high contrast. The requester's Ambient tab has these options:

Color: This area has sliders (0-255) for choosing the component colors of the light: 0 to 255 levels each of Red, Green, Blue.

Intensity (0-100%)

Limits: None

Menu Item: Fog
Keyboard: None



Brief: Specify Global Fog attributes

This menu item allows you to specify the attributes and position of Global Fog. This will not be used in the render unless the global permission flag is ON (in Render Settings). This menu item opens the Fog Settings requester.

The gadgets in this window are:

Render Fog (check box): This toggles global fog rendering on or off. You can set up and text Fog, and then turn it off till it's time for your final rendering.

Color (Slider and numeric entry box for each of Red, Green, and Blue): The color for the fog.

Near: This is the near position of the global fog. Objects closer than this will be out of the fog.

Far: This is the far position of the fog. Objects farther than this will be completely enveloped in the fog.

Strength: This is the fog's Strength. It should almost always be left at 1000.

Fog is conceptually very easy. The Near parameter is usually set to 0 and the Far setting is the distance you want to see. If you're using the camera, the camera's position is at 0.

The virtual camera position is the distance from the origin that you have specified as Observer Position. You can find the position of the virtual camera by asking for the Observer Position (RMB on the Zoom gadgets in the Editor).

Fog is easiest to set up when using a camera. Global fog is effective in allowing objects to appear out of the fog and disappear into the fog as they move through your space.

Another use for fog is as a traditional Depth Cue. To use fog in this manner, choose your Render Settings as usual, but do not use local or global lights. Set Ambient to 1.0. Then turn on the global fog, using a suitable range, but with a black color. The image will render, complete with textures if desired, but the polygons will be darker, the farther they are from the viewer.

Limits: None

Menu Item: Rounding
Keyboard: None

Brief: Specify the Rounding variable This menu item allows you to change the Rounding variable - it's specified in the Rounding Control requester.

This is not a shading type. Rounding is a power function. Usable range is any positive number.

Menu Item: Background
Keyboard: <F8>

Brief: Select background Texture List

This menu item opens the Available Texture Lists selection window. If you already have a background chosen its name will be selected. Here you can alter or choose the Texture List you want for a background. See the section on Texture Lists for more information.

Backgrounds are the first layer in the image. They are always behind all polygons in the drawing. Lessening the strength of the background effectively blends Color 0 into the image. The background can use any procedural or bitmap texture in the Texture List and obeys all Time line and Strength instructions so it can use multiple composited images or fade one from the other. Backgrounds can be bump maps, but are painted as though the light always comes from the upper right of the screen. Backgrounds can use animation sequences or animated procedurals. For the procedurals, the background is a virtual plane in space from 0 to 40000 in the XZ plane. The X runs left to right, and Z runs up and down. Backgrounds are always painted as though projected from the Y axis.

TIP: If you're creating an animation using processing in the background texture list and the image is the same throughout the animation, it is usually faster to render it once, save it and then load it as a single image texture list. This way the compositing is only done once instead of once per frame.

Limits: None

Menu Item: Foreground
Keyboard: <Ctrl> <F8>

Brief: Select foreground Texture List

This menu item opens the Available Texture Lists selection requester. If you already have a foreground chosen it will be selected. Here you can edit or choose the list you want. See the section on Texture Lists. Foregrounds are Texture List items that are composited as a front image just before the overlay. They are optimized so no rendering takes place where a foreground covers the render. Lessening the strength of a foreground reveals, not the render, but Color 0 under the image. This may be used to advantage in genlock and blue screen work.

The foreground can use any procedural or bitmap in the Texture List and obeys all Time line and Strength instructions so you can use multiple composited images or fade one from the other. Foregrounds cannot be bump maps unless the bump map is saved with a Color 0 area and used as a normal image. Foregrounds can use animation sequences or animated procedurals. For the procedurals, the foreground is a virtual plane in space from 0 to 40000 in the XZ plane. They are always painted as though projected from the Y axis.

Important: The foreground should always be used in Decal or Genlock mode or the Color 0 areas in bitmaps will not be transparent.

If you're creating an animation using processing in the foreground texture list and the image is the same throughout the animation, it is usually faster to render it once, save it, and then load it as a single image texture list. This way the compositing is only done once instead of once per frame.

Limits: Foregrounds should always be in Decal or Genlock mode when using bitmaps. Bumpmaps are not permitted.

Menu Item: Overlay
Keyboard: <Shift> <F8>

Brief: Select overlay Texture List

This menu item opens the Available Texture Lists selection window. If you have already chosen an overlay, it will be selected. Here you can alter or choose the list you want as an overlay. See the section on Texture Lists for more information.

Overlays are Texture Lists that are composited as a final stage into the image. Their strengths should always be less than 1.0 or the render under will be complete concealed.

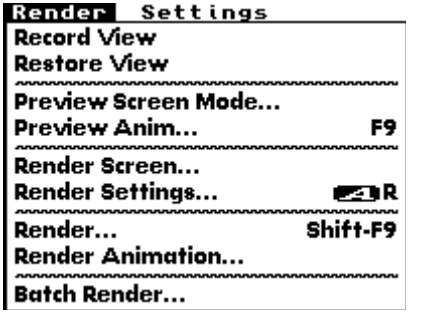
The overlay can use any procedural or bitmap in the Texture List and obeys all Time line and Strength instructions so they can use multiple composited images or fade one from the other. Overlays can be bump maps but the light always comes from the upper right of the screen. Overlays can use animation sequences or animated procedurals.

For procedurals, the overlay is a virtual plane in space from 0 to 40000 in the XZ plane. The X runs left to right, and Z runs up and down. They are always painted as though projected from the Y axis.

If you are creating an animation using processing in the overlay texture list and the image is the same throughout the animation, it is usually faster to render it once, save it, and then load it as a single image texture list. This way the compositing is only done once instead of once per frame.

Limits: You cannot use this menu item during facing or deform editing.

The Render Menu



Menu Item: Record View
Keyboard: None

Brief: "Snapshot" the Editor's view, so you can come back to it easily

While working, it's often useful to take a look at your objects and polygons from another direction - isometric, perhaps, or just scoot the Editor view over a bit to see what's out there. If you use the Render menu item Record View before you change it, you can come back to the same view easily and instantly with Restore View, the next option.

Limits: None

Menu Item: Restore View
Keyboard: None

Brief: To revert to a previously "snapshot" Editor's view

While working, it's often useful to take a look at your objects and polygons from another direction - isometric, perhaps, or just scoot the Editor view over a bit to see what's out there. If you use the Render menu item Record View before you change it, you can come back to the same view easily and instantly with this menu item, Restore View.

Limits: None

Menu Item: Preview Screen Mode
Keyboard: None

Brief: To set up the default screen settings for Preview

This menu option is where you tell Aladdin 4D what Amiga screen mode you'd like to use for Preview. The option opens a standard Amiga Display Database requester in which you can select a mode.

Depending on which model Amiga you have and which monitor drivers you have installed into the Devs:Monitors drawer, and whether or not you have an AGA Amiga or one of the many optional third-party graphics cards, you may see only a few or many screen modes in this requester, from which you can select one.

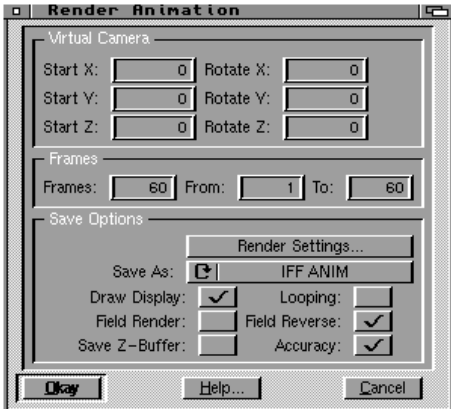
Limits: None.

Menu Item: Preview Anim
Keyboard: <F9>

Brief: To check your placements with a quick wireframe animation

This option lets you take a look at your objects in motion, without waiting for a full render to take place. The objects are shown in wireframe on a 1-bit plane screen (see Preview Screen Mode), and their positions are calculated "on the fly" from the Editor view (if you don't have a camera object defined). You can save the Preview Anim to disk for further study, using the Save As gadget in the Render Animation requester that opens.

Preview operates as fast as it can, considering the number and complexity of objects in your drawing. If it has to draw many, many vectors, it'll step through your animation much more slowly than if it only has to calculate a few. If you need to see the motion at a more natural speed, save the animation, and then view it with any standard Amiga animation viewer.



The Render Animation requester has these gadgets:

Virtual Camera (the Editor's view, unless specified otherwise here)

Start X, Y, and Z: Where the virtual camera's view starts

Rotate X, Y, and Z: The rotation of the virtual camera during the animation.

Frames: The number of frames to preview, and which ones...

From: The first frame to preview

To: The last frame to preview

The requester offers a Render Settings button at the bottom, so you can set up Globals and permissions before generating the Preview Anim.

Limits: None.

Menu Item: Render Screen
Keyboard: None

Brief: Establish your preferences for the Render screen mode

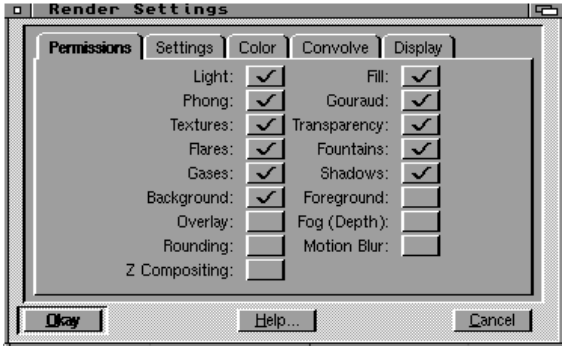
This menu item allows you to choose a new screen, or display mode for the program's rendering.

When you select this menu item it opens the Render / Screen Mode window. Here you can select new resolutions, display devices, number of colors, etc. This is where you tell Aladdin 4D to support hardware display boards.

The settings in this requester don't change the screen on which Aladdin 4D's user interface appears. They only tell the program what resolution you want pictures rendered in. You can change these settings any time - useful in making quick tests.

Limits: None

Menu Item: Render Settings
Keyboard: <R>



Brief: Select settings for the screen to be rendered

This menu item gives you access to the global "permission" flags for the render~~.~~ _These are selections that allow you to proof your render without some of the options you want for the final render~~.~~ _For further narrative discussion of the elements of the Render Settings, look into the opening section.

Selecting the Render Settings menu item opens the Edit Render Settings menu~~.~~ _The gadgets in the requester are:

~~(TAB)~~ Permissions Tab

~~/// TEK: You have *got* to be kidding me… Open up the old Aladdin manual to page 163 and notice how much more information there is about these options there than there is here. Sheesh. That's it, I'm not reading any more.~~

Light: Whether lights are ON or OFF

Phong: Enable Phong shading if ON

Textures: Render (or don't) Aladdin 4D's Texture Lists~~.~~ _Turning this off greatly speeds test rendering during development of your animation.

Flares: Enable Flares compositing by turning this option ON (Phong shading must also be turned ON).

Gases: Turn ON gases.

Background: Turn ON rendering of backgrounds (when this is checked, it's on)

Overlay: When checked, Overlay rendering is on.

Rounding: Turns on Rounding when checked~~.~~ _Rounding is configured in the Environment / Rounding menu.

Z Compositing: Turns on Z Compositing.

Fill: Fill is on when this is checked.

Gouraud: Gouraud shading is enabled when this is on.

Transparency: Click this to turn on Transparency calculations

Fountains: Fountains work when this is checked~~.~~ _Otherwise, they don't render.

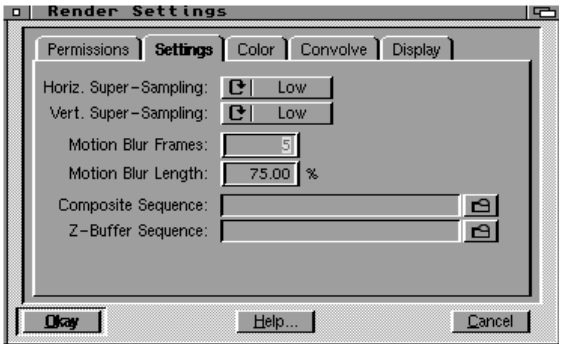
Shadows: Turn on shadows. Shadow rendering takes LOTS of rendering time, so don't turn this on till you need it.

Foreground: Enables Foregrounds in the rendering.

Fog (Depth): This option turns on Global Fog.

Motion Blur: Motion Blur is only useful in screen resolutions that are friendly to video, and only if you're creating animated sequences of rendered frames. This check box turns Motion Blur on. Generally, leave this off unless you're outputting to video. When on, it creates extra smooth motion by simulating the slow shutter speeds common to most video recording equipment. Also see Motion Blur Frames and Motion Blur Length in the Settings tab.

(TAB) Settings Tab



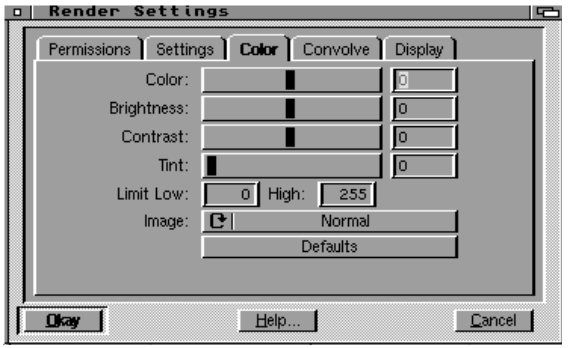
Horiz. Super-Sampling (rotary gadgetcycle gadget): This gadget can read None, Low, Medium, or High. It's the control for antialiasing (global) in the horizontal direction.

Vert. Super-Sampling (rotary gadgetcycle gadget): This gadget, also, can read None, Low, Medium, or High. It's the control for antialiasing (global) in the vertical direction.
"High" antialias settings render 16 rays per pixel. The "Medium" antialias setting renders 9 rays per pixel.

Motion Blur Frames: Motion Blur works by rough-rendering a range of slightly different positions of the objects and compositing them as one frame. This is the number of frames Motion Blur will use to calculate its effect. Higher numbers that 5 are probably a waste, and lower than 3 will make a very subtle effect. The default is 5 - a good setting for most purposes.

Motion Blur Length: This is the percentage length of the motion blur. The default is 75 percent, a good setting for most purposes.

~~(TAB)~~ Color Tab



Slider and numeric entry box for each of Color, Brightness, Contrast, and Tint~~.~~ These apply global controls to the rendering operation.

~~@~~ Color: Adjusts the color saturation of the image. ~~/// ??? 0507 /j~~

Brightness: The overall brightness of a scene can be altered with this option.

Contrast: The contrast of a picture is a measure of how rapidly its dark areas transition to its light areas - it's the slope of the gamma curve, a graph of darkest to lightest densities in the image~~.~~ Contrast is controlled by the lighting scheme, especially the amount of Ambient present, as well as the directionality of the light sources~~.~~ Broad sources produce less contrast that point sources or conic light sources ("spotlights" in the lighting trade)~~.~~ This option gives you an additional, simple control over the contrast of the scene.

Tint (Slider): The overall color balance of a scene can be corrected or altered with this slider~~.~~ Use it to impose or correct a color cast.

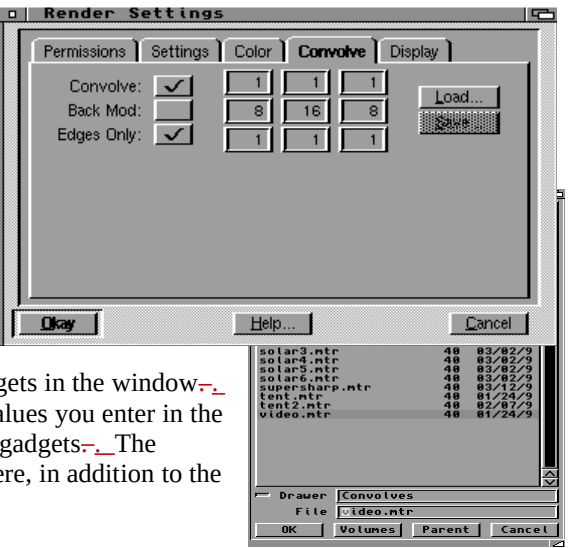
Limit Low: This tells Aladdin 4D to limit the lowest - that is, darkest - color of your scene to the value you specify~~.~~ The value can be 0 to 255, but as this approaches 255, indeed as it moves past 100 or so your scene will look ever more washed out - a useful effect perhaps for some purposes~~.~~ This and the Limit High option are really intended to permit you to restrict the tones in your picture to accommodate the limits of certain displays - say, NTSC.

Limit High: This tells Aladdin 4D the maximum value you want to use for "white" in the picture~~.~~ See Limit Low.

Image (~~rotary gadget~~cycle gadget): This selects whether you want the image rendered as Normal or Negative~~.~~ If you want a picture to look like a photographic negative, turn this ON.

Defaults (button): Restores all Aladdin 4D Render Settings defaults.

~~(TAB)~~ Convolve Tab



Convolve: Convolve operations won't happen till you turn this check box on~~.~~ If this gadget is selected, the convolution you have entered in the number gadgets will be applied to the image during the render~~.~~ If you are not familiar with this operation, understanding is simple, and the option's power is immense~~.~~ The program will look at the color of a grid of nine pixels, with the one currently being painted in the center of eight others.

This arrangement is reflected in the nine Convolve gadgets in the window~~.~~ The color values in these pixels are multiplied by the values you enter in the gadgets, then divided by the sum of the numbers in the gadgets~~.~~ The process is repeated for each pixel - lots of math to do here, in addition to the

render.. If you have "1" in the center gadget, and 0's in all the others, the result is no change in the image.. All 1's will blur the image.

Many effects may be achieved by varying the numbers used.. Several examples are provided on the distribution disk and may be loaded.. Load/Save only loads and saves the convolution matrix of numbers, not the other settings in the image window.

You can save the convolve pattern, and load them back in for further use or refinement.. Only the convolution numeric matrix is saved, not the other settings.

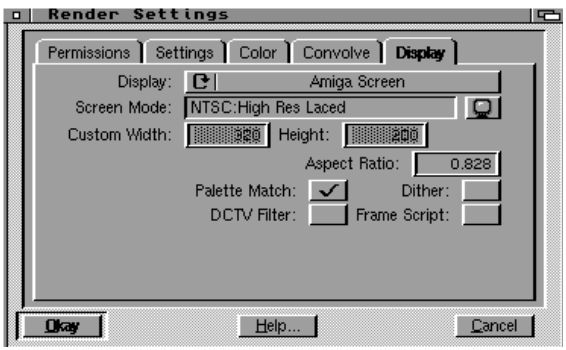
Back Mod: If ON, the convolve will be applied to the background.

Edges Only: The Aladdin 4D Convolve is able to do edge detection - not just a color threshold.. It can tell if polygons participate in the same shared list at edges, and it can detect when polygon/background edges have been hit.. If you have Edges selected, the convolve will occur only at the edges of polyspolygons and backgrounds, and polyspolygons that are not shaded together.

(TAB) Display Tab

The Display tab is where you establish what type of display you have, whether it's an Amiga screen, or a third-party display board.

Display (rotary gadgetcycle gadget): Rotates among the following options:



- Amiga Screen – an Amiga screen of any ——resolution, including AGA modes.
- DCTV - The DCTV display device.
- OpalVision - third-party display card.
- Resolver - third-party display card.
- Retina - third-party display card.
- Video Toaster - The NewTek Video Toaster, with or without the Video Flyer card.. Aladdin 4D can render directly to the Toaster or Flyer, if it's present and if the Toaster or Flyer software is also running.
- None (Render to disk).. When you select this option, Aladdin 4D will not display the rendered image as it creates it, but will store it to a path and filename you supply in a standard system requester.

Screen Mode: This button lets you pick a display mode (say, Ham-8 if you have an AGA-equipped Amiga) from the display database.

Custom Width: Specifies the width of the rendered image in pixels.. This should be a multiple of 16.

Custom Height: Specifies the height of the rendered image in pixels. This should be a multiple of 16.

Aspect Ratio: The default is 0.868, the normal Amiga aspect ratio for most normal Amiga monitors.. Aspect ratio can be affected by the dials and controls on the monitors themselves, so there's no scientific

way to do this, except to test it. Draw a perfect circle or square on the screen, and fiddle with this setting until it comes out perfect on the display you intend. Of course, if you then change the monitor, or its knobs, you'll have to do the setup testing again.

The Aspect Ratio box specify the RENDER screen's aspect ratio, and can be independent of the setting for Editor and Preview displays. Also, it does not have to match the aspect ratio for the screen mode you choose with the Render Screen option.

Video displays vary widely. Frequently the pixels that compose the screen are not exactly square, particularly with Amiga displays. So, if you're creating images for use on non-Amigas, you'll need to adjust Aladdin 4D's Aspect Ratio specification to the aspect ratio for the target display. Aspect Ratio is the width of the image divided by the height. In most video applications, this is approximately 4/3, or 4:3, assuming the pixels are square. For other purposes, it might be considerably different. Computer monitors are not necessarily the same, even if driven by the same brand of computer.

So, test, test, test, to get the aspect ratio you render to match the pixel aspect ratio of the system you will be using to view the image.

For a typical Amiga monitor, the default of 0.8281 will cause a circle in space to appear as a circle on the monitor. Not all Amiga monitors are the same, however, so if this doesn't work for you, don't be alarmed. In addition, most monitors have ~~knobs which~~knobs that change their horizontal and vertical screen size. So, there isn't much predictability to the process.

Try the default setting first, and if it doesn't show round-for-round, square-for-square on your monitor, then consider changing the program's Pixel Aspect Ratio settings or twiddle the H-size and V-size knobs (if your monitor has them) or both.

For desktop publishing and for output to most film recorders, you should specify an aspect ratio of 1.0 - that is, square pixels - and an image size to meet the need of the output resolution you will be doing. Desktop publishing applications are resolution hungry, so plan to be generous with the pixels to get sharply printed results. If you're going to print, also be careful of the overall density of the picture. Mainly, don't make the shadows too deep, and if there is important detail in shadow areas, think in terms of brightening them to help out the reproduction.

Palette Match: (~~e~~check box) This is available only in the Amiga's special HAM modes. It does not apply to any other display mode. If selected, it forces the color desired to match one in the palette. This effectively prevents any HAM fringing, but may cause an undesirable color placement. See the tutorials for best use. This requires the fill mode to be on. This option works as well if you are running an AGA chipset machine and are in HAM8 mode. ~~/// ???~~

DCTV Filter: Turn this on if you're using DCTV. It "filters" certain types of display information so DCTV's display routines will make a better picture.

Dither: (check box) This is available in regular Amiga display modes only. If selected, an appropriate dither for the mode in use will be applied. This requires the fill mode to be on. ~~/// ???~~

Frame Script: (check box) If you are producing a single frame animation, this allows a frame script to execute after the completion of each frame. The frame script exists in your s: directory. An example has been provided. Using this option you can do image processing, etc. on each frame as it is produced. ~~/// ???~~

~~/// These are missing compared to version al30 0507 ??? /j~~
~~@ (rotary gadget)~~
~~@ IFF ANIM~~

@ Individual 24-bit ILBMs
@ Individual ILBMs
@ Draw Display: (check box)
@ Field Render: Field rendering simulates the action of video recording
@ equipment by rendering a picture in two interleaved fields rather than a
@ single, non-interlaced frame. This turns Field Render on. Leave it off
@ unless you need it - it's only for animation output to video.

Save Z-Buffer: (Check box) /// ???
/// Added ability to save Z buffer information to a ZBuffered file.

/// This is missing, compared to v al030 0507 /j
@ Looping: (check box)
@ Field Reverse: (check box)
@ Accuracy: (check box)

Menu Item: Render
Keyboard: None

Brief: Tell the program to render a single picture.

Frame (slider): This gadget lets you select which frame of an animation you want rendered. If you're creating a single picture, leave this at 1.

Save As (rotary gadgetcycle gadget): Here's where you tell Aladdin 4D what kind of file to save. The choices are:

- None: This means don't save the picture to a file, just render it. Useful for checking your work in lower resolutions before you commit to a full render.
- Iff-ILBM: Save a standard Amiga IFF picture in the format determined by the Render Screen menu (Ham, AGA, Hi-res, Lace, whatever).
- 24-Bit ILBM: Saves the picture in 24-bit form, whether or not you have a 24-bit display card in your Amiga.

Render Settings (button): Brings up Render Settings requester for last-minute changes. If you open it from here, when you close the Render Settings requester, you'll come back to the Render requester.

Menu Item: Render Animation
Keyboard: None

Brief: Render an animation in a variety of ways.

This option brings up the Render Animation requester. Its options are:

Virtual Camera:

Start X, Y, and Z:

Rotate X, Y, and Z:

Frames

The number of frames you want in the animation~~.~~. Video plays at approximately 30 frames per second, so if you want a one-second animation, select 30~~.~~. It's a good idea to put some extra frames at the beginning and end of an animation intended for video, to make the editing less critical.

From: First frame to start~~.~~. Use this to test renders in the middle of your animation, if you need to.

To: Final frame of the render~~.~~. This lets you test render only a chunk of the sequence.

Save Options

Render Settings (button): Brings up Render Settings requester.

Save As: This ~~rotary gadget~~cycle gadget steps through the available Save options~~.~~. Aladdin 4D cannot create realtime animation to the Amiga screen, so the animated sequence has to be saved in some form or other and then viewed with an external program~~.~~. The Save possibilities are:

IFF Anim: A standard Amiga animation format~~.~~. If you've selected an AGA Render Screen (see the menu option Render Screen), this will be in a matching format, such as Ham-8.

Individual 24-bit ILBMs: This will save the sequence as numbered individual frames to your hard disk~~.~~. These frames can be quite large, so your available hard drive space should be plentiful if you're rendering many frames~~.~~. These are 24-bit pictures, too~~.~~. In high resolutions, expect very large files.

Individual ILBMs: Saves a sequence of rendered ILBM images.

Individual Framestores: Saves a sequence of Video Toaster-compatible framestore images.

Draw Display: If this is ON, Aladdin 4D will draw each frame to the Amiga screen as it creates them~~.~~. If OFF, the pictures are rendered to disk without being shown on the screen.

Field Render: Field rendering is for use in creating animation intended for playback in video~~.~~. Turn this ON to have Aladdin 4D render field wise, instead of frame wise - that is, alternate scanlines of two, not-necessarily-identical interlaced fields, instead of a single two-field frame~~.~~. The result is smoother motion, and a much closer simulation of "real" video footage~~.~~. The penalty in added render time is not bad - not double, at least.

Field Render: Field rendering simulates the action of video recording equipment by rendering a picture in two interleaved fields rather than a single, non-interlaced frame. This turns Field Render on. Leave it off unless you need it - it's only for animation output to video.

Field Reverse: Reverses the even and odd fields output when field rendering.

Save Z-Buffer: (check box) When selected, and saving a sequence of individual 24-bit ILBMs, the Z-Buffer information for each frame is saved to disk as well. The Z-Buffer information is saved to a file with the same name as the corresponding frame, with an extension of ".zb". This information can be used for 3D compositing in Aladdin 4D or other packages.

~~Save Z-buffer: (check box) /// ??? 0508 /j~~

Looping: (check box) When generating an IFF ANIM file, this option causes a looping animation to be created.

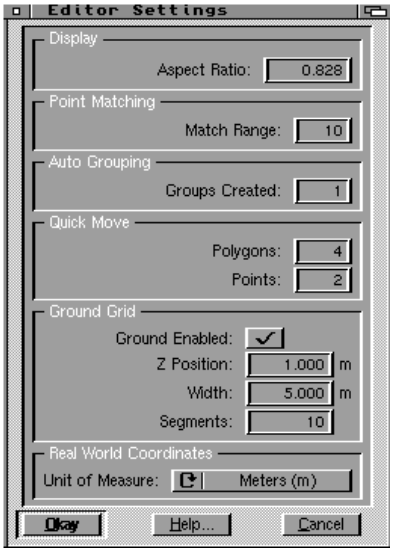
~~Field Reverse: (check box)~~

Accuracy: (check box) Normally this is always checked – it enables a higher quality rendering method.

~~Batch Render (Requester: Bach Render Control)~~
~~For each of 10 Numbered areas 1 through 10:~~
~~Drawing (filename)~~
~~Output (filename)~~
~~GADGETS: Remove, Clear All, Cancel, About, Perform, Accept~~

The Settings Menu

Menu Item: Editor Settings
Keyboard: None



Brief: This item lets you establish some basic settings for the Editor, including your choice of coordinates systems and an adjustment for the aspect ratio of the specific monitor you're using.~~-~~. The option opens the Editor Settings requester.

Aspect Ratio: The default is 0.868, the normal Amiga aspect ratio for most normal Amiga monitors.

(Point Matching) Match Range: This menu item allows you to change the distance over which the Match Points tool will find matching points.

The number in the Match Range gadget specifies the distance that the Match Points tool will use to recognize two points as being the same in space.~~-~~. See the section on the Match Points tool.

(Auto Grouping) Groups Created: This panel gives you control over the way polygons are grouped during lathing and similar operations.~~-~~. By changing the number in the Groups Created gadget, you control of groups that are made during these operations.~~-~~. The polygons are staggered in their group numbers during creation.

Quick Move panel

This panel allows you to manually enter the parameters that Quick Move will use when painting polygons.

Polygons: This gadget contains the number of polygons that will participate in the display.~~-~~. At a setting of 4, the program will display polygon numbers 1, 5, 9, 13, 17, etc.

Points: This gadget displays the number of points in the participating ~~polys~~polygons.~~-~~. The minimum number of points is 2.~~-~~. You can also control Quick Move interactively through the <.> period and <,> comma keys.~~-~~. When shifted, they control the number of points; unshifted they control the number of ~~polys~~polygons.~~-~~. See the section on the Quick Move tool.

This menu is also where you set the Editor's ground grid, and whether you want to use one.~~-~~. A can make it easier to visualize your placement of objects, and facilitate exact relationships among the objects placed.~~-~~. If you set Width to a value greater than the distance of the virtual camera (found in the observer position window, called "Image Distance"), the Editor will automatically go into full 3D clipping mode.~~-~~. This will result in slower Editor update, but allows polygons (and the grid) to go behind the camera position in the Editor.

Ground Enabled: By default, the ground grid display is ON.~~-~~. You can turn it off here, if you want to.

Z Position: This sets the coordinates for where the ground grid will be placed on the Z axis.

Width: The width of the grid.

Segments: The number of divisions you want for the ground grid.

Real World Coordinates: Allows you to select the unit of measure that the coordinates will be displayed in.

Menu Item: Coordinates
Keyboard: None

Brief: Toggle display of coordinates

Menu Item: Axis
Keyboard: None

Brief: Toggle display of axis

This menu item turns the display of the axis on and off. It is sometimes convenient to see the display without the axis on screen.

Limits: None

Menu Item: Auto Quick
Keyboard: None

Brief: Allow auto Quick Move

This menu item, if flagged, allows the program to automatically go into Quick Move mode when you rotate your view to use certain real-time tools. Auto Quick will be engaged if you have more than 50 or so polygons in your drawing. This is helpful in reaching a new view angle quickly. If you want to use tools sensitive to Auto Quick, you can turn this off. With it off, you can enter and exit Quick Move by pressing the <q> key or by clicking the Quick Move gadget.

Menu Item: Save Icons
Keyboard: None

Brief: Specify whether icons are saved

This menu item when checkmarked causes the program to save icons for some of its file types. If not flagged, the icons are not saved. Icons make it easy to navigate through your drawings using the Amiga's Workbench. You can simply double-click a saved drawing's icon to have the Amiga load Aladdin 4D and then the drawing.

Limits: None

Menu Item: Toolbar Window
Keyboard: None

Brief: Toggle the Toolbar Window.

This item turns the Toolbar window off or on.

Limits: None

Menu Item: Ext. Toolbar Window
Keyboard: None

Brief: Toggle the External ("Ext.") Toolbar Window.

This item turns the External Tools window off or on.

Menu Item: Save Settings
Keyboard: None

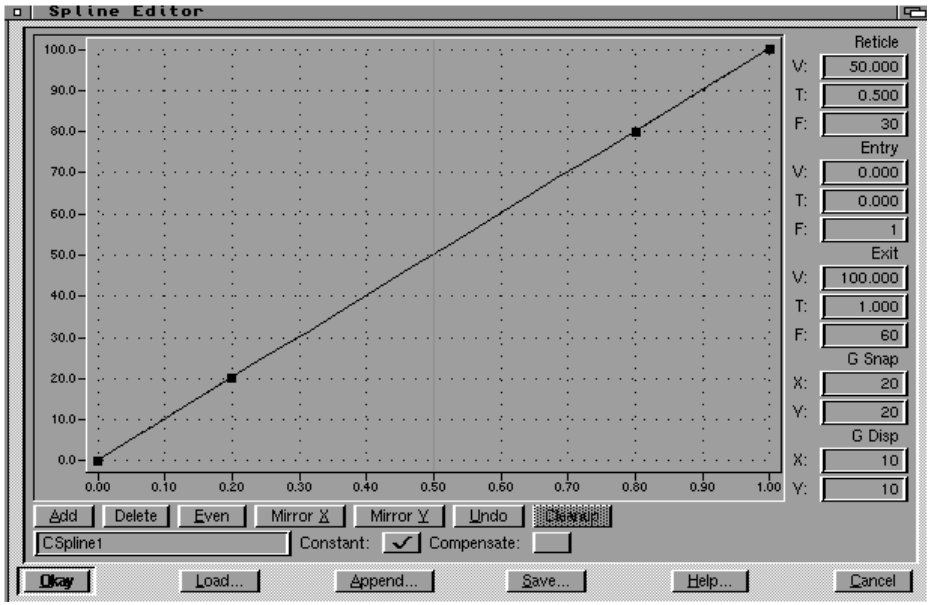
Brief: Record current Editor Settings as defaults.

This item stores the current Editor settings as the Aladdin 4D defaults, so the next time the program is run it will come up as you had it set when you last used this menu option. Aladdin 4D stores render and preview screen settings with each saved drawing, but not Editor screen settings. This stores your preferences in a "defaults" file on your hard disk, overwriting the previous file, if present.

Limits: None

CSpline Editor

Within Aladdin 4D many parameters can be turned over to CSplines - that is, "Control Splines" for non-linear operation during animation. These CSplines establish mathematically defined curves which the program uses to interpolate values for a given frame of animation, based on the limits = Entry and Exit, usually - that you supply.



The CSpline Editor is available from most requesters by clicking a CSpline gadget. The gadgets and features in the CSpline Editor are extensive, and some of them open requesters of their own.

When you click the gadget, you're given the opportunity to pick a previously created, named CSpline from a list of what's available. If you're working on a new drawing, there aren't any active CSplines, however, so the only option that's available is "New" which makes a new one. Once there are CSplines to choose from, you can choose Edit - alter an existing CSpline - or Free, which releases the CSpline control from the polygons assigned to it.

The main area of the CSpline Editor shows the CSpline curve and its control points. As you move one of the control points (they're about a third the way over from the end points), the program performs some math to smoothly arrange the line, based on the positions of its anchor and control points. You can move the anchors, too, if you need to.

Along the left side of the edit window, the percentage rule usually shows from 0.0 to 100.0 percent. If you are editing a function that is known, however, like the angle of a rotation, the rule will display the actual numbers.

Across the bottom of the edit window, the TIME rule always shows from 0.0 to 1.0. This is the position in time for the function the CSpline is to control. It may be the entire animation or just a part of it, depending on the member's assigned time slice. The Frame gadget displays the actual frame numbers and reflect partial time spans properly.

To make symmetrical changes to the spline across junctions in segments, hold down <Ctrl> while moving a control point. You can also hold down <Ctrl> when moving an anchor point, so as not to change the position of the control points.

~~/// ??? Is this changed ??? 0107 /j~~

Values: The gadgets on the right side of the requester display the actual percentage values at various points along the CSpline. The V: gadget displays the "Value" of the current position. Time displays the current time within the time slice being edited, and Frame displays the frame number. All of these values are repeated for each of the possibilities - Entry, Exit, and Reticle positions. The "Reticle" is the vertical bar in the middle of the curve. You can move it around to check values for a given position along the curve.

Reticle: At reticle

Entry: At first point of the active segment (or whole CSpline)

Exit: At last point of active segment (or whole CSpline)

If you prefer - or if you have some specific value you want for a given position, you can simply enter numbers into any of these gadgets, and the graphic display will adjust itself accordingly.

G.Snap is the Grid Snap option, and G.Disp displays (or not) the grid. G.Disp controls the grid displayed. This may be the same as the snap, or different. Use small numbers when zoomed out, larger numbers when zoomed in. The percentage and time rules correspond to the values you set here. X controls the time and Y controls the percentage

In G. Snap X and Y, if you specify, for example, an X snap of 2, this means the points you move will automatically jump to either 0.0, 0.5, or 1.0 in time. The Y snap controls the percentage portion of the grid. Snap is always ON. Use values of 0, if you want no snap.

CSplines need not be a single curve; however each segment of a CSpline will be treated as a spline curve. A spline actually doesn't have to be curved, a straight line is acceptable, and for some purposes might prove useful in designing a complex motion. To place more curves into the window for editing, use the Add button at the bottom. Previously saved CSplines can be incorporated, using the Append option, which offers more control over how the additional CSpline is to be treated on import.

Add: Adds a new segment to the end of the CSpline if no segment is active. If a segment is active, it is added just after the active segment. This is not undoable.

Delete: Deletes the active segment. This is not undoable. If you inadvertently hit this gadget and don't want to lose the original CSpline, click Cancel. The Editor will close, and the original CSpline (without any changes you may have made) will be intact.

Even: Spaces all segments of the CSpline so they occupy the same amount of time and are continuous.

Mirror X: Mirrors the current CSpline along the X axis

Mirror Y: Mirrors the current CSpline along the Y axis

Undo: Undoes the last action (some actions cannot be undone, however)

Cleanup: Removes the CSpline you have been working on and resets to default settings and a single linear segment. If you inadvertently click this gadget and don't want to lose the original CSpline, select Cancel. The Editor will close and the original CSpline (without any changes you may have made) will remain intact.

Spline Name: The gadget near the bottom shows the name of the CSpline. You can edit this to your liking. The name is retained with the drawing and saved if the CSpline is saved. Use as many

characters as necessary, maximum of 40. This, like all gadgets in Aladdin 4D, is a standard Amiga Intuition gadget and requires a <Return> before any change is recorded.

Constant: This is not saved with the CSpline. Instead, when the Editor opens a CSpline, if the anchor points are at 0.0 and 1.0, and the segments (if more than one) are joined, the Editor selects this gadget for you. If you turn it OFF, you may move the end points away from 0.0 and 1.0, and you may have a non-continuous CSpline by moving segment points away from each other. If you have a non-continuous CSpline and hit this gadget, the CSpline will be made continuous. This is not undoable! If you inadvertently click this gadget and don't want to lose the original CSpline, select Cancel. The CSpline Editor will close, and the original CSpline (without any changes you may have made) will be intact.

Compensate: If you have this gadget selected, when you move an Endpoint for a spline segment along the time axis (left and right), all other points in all other segments of the spline will be compressed or expanded relative to their distance from the point you're moving.

Load: Loads an existing CSpline from disk. The CSpline does not replace the one being edited, but is added to the list. Editing the loaded CSpline will not affect the CSpline on disk unless you save over it. It is quite possible to load a CSpline multiple times, and it will be placed in the list multiple times by the same name. Confusingly - but do it if you wish.

Save: Saves the current CSpline. You will be asked for a filename each time you save. It is recommend you name the CSpline the same as the project's filename, but it does not have to be the same. It is also recommend you name the CSpline without spaces to avoid confusion in filenames. Use of the underscore character is the usual method: "My CSpline" would be "My_CSpline".

Append: This gadget opens a requester where you can set defaults for appending a spline that has been previously saved. In this way you can build complex shapes from multiple CSplines. There are three settings in the Append requester:

Time Allocate: Ranges from 0.0 to 1.0. This is the amount of the time displayed that the appended CSpline will occupy. The current CSpline will be reduced to fit the balance of the time line. If you use 0.5, the current CSpline will be reduced to the first 0.5 of time, and the appended CSpline will get the last 0.5. If you use 0.25, the current CSpline will be reduced to the first 0.75 of time and the appended CSpline will get the last 0.25.

Scale Value: Allows you to scale the value component (vertical) of the appended CSpline. A setting of 1.0 means the appended CSpline will come in at its saved values. At 0.5, it will come in at half its saved value range. Negative numbers invert the spline.

Offset Value: Allows you to position the appended CSpline along the value axis (vertically). A value of 0.5 positions the appended CSpline at its saved position.

Accept: Opens Aladdin 4D's file requester so you can select a spline to append in the manner you've set.

Important: These values are NOT saved. You must edit them each time you append a spline.

CSpline Editor Keyboard Controls

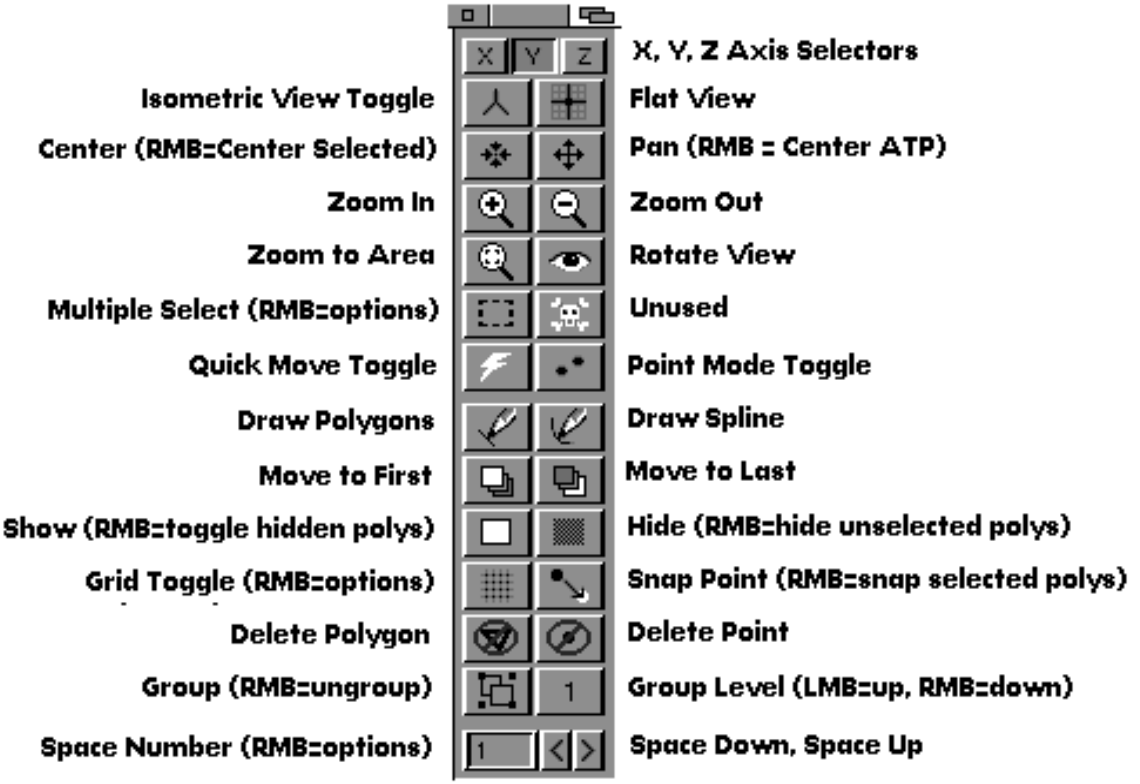
In addition to the gadgets just described, some keys are useful in the CSpline Editor. They are:

<Ctrl>: If held down when editing the position of a control point, the opposing control point will take on the same magnitude and an opposing position making the slope of the opposite sides of the junction the same. Used with CSplines with multiple segments.

<Cursor>: These keys control the page position allowing you to pan around when zoomed in.

<Esc>: Undoes position changes if used before clicking empty space in the Editor (spline turns black) or moving another point.

Gadgets (Internal)



Name: Isometric
Access: LMB
Secondary: None
Keyboard: <F7>

Brief: View the current drawing in wireframe from a parallel (engineer's) view.

In isometric mode all lines that are parallel in space are drawn parallel on screen. All references to perspective are eliminated. This mode is very useful for exact placement of objects relative to others regardless of their distance from the viewer.

Limits: Isometric mode is not available in the render screens. It is, however, available in Preview.

Name: QuickMove
Access: LMB

Secondary: None
Keyboard: Q

Brief: Speed up the Editor and preview display

While QuickMove is on, only selected portions of each polygon are refreshed. resulting in a dramatic increase in screen updating. Control is provided through the <.> (period) and <,> (comma) keys. When in QuickMove, the period key decreases the number of sides that are refreshed, and the comma key increases. Using these keys with the <Shift> key will increase or decrease the number of polyspolygons that are refreshed. In Preview, use the <Q> key to enter and exit QuickMove, and the period and comma keys to control the display. You can also set the QuickMove values manually.

Limits: None

Name: Active Axis Gadgets
Access: LMB
Secondary: None
Keyboard: None

Brief: Specify an Active Axis.

The axis that you specify as "active" controls the way the program sets up the view and controls the behavior of some of the tools. When you select the Flat View gadget, the view is presented as though you are at the end of the selected axis. For example if the Y axis is active, using the Flat View gadget will present the view as though you are at the end of the Y axis, with the XZ plane flat to the screen. The Freehand tools reference the Active Axis to know which plane to draw in. Some other tools (including externals, see their about files) may reference the Active Axis to know which axis to deal with. And finally, when you select and move polyspolygons, the Active Axis determines which plane they move in. You can view from one axis and move in any other.

Limits: Only one axis may be selected at any time.

Name: FreeHand (Polygon)
Access: LMB or <TAB> key when a polygon is selected
Secondary: RMB (Manual Entry)
Keyboard: None

Brief: Draw freehand polygons.

Clicking this gadget allows you to draw polygons. The new polygon will be started at the current Attach Point. Pressing and holding the left mouse button will drag a new point out from the last one. Tapping the <TAB> key will leave this point at its current position and create a new point. This may be done any number of times to create a polygon of any number of sides, each of any dimension. The plane that the new point moves in is controlled by the Active Axis. If Y is selected, the movement of the point will be in the XZ plane. All points of a multi sided polygon should be co-planar, or the polygon will not render properly.

If you select an existing polygon, then tap the <TAB> key, a new point will be created in the selected polygon and you will enter FreeHand mode. (If you want only one additional point, click the right mouse button on the Delete Point gadget as explained later) Additional points may be created by additional taps on the <TAB> key or additional right mouse button clicks on the Delete Point gadget. The point is not

attached to the mouse pointer. Think of the mouse pointer as simply indicating a direction and amount of movement for the point. If you select the FreeHand gadget with the right mouse button, you enter Manual Mode where you can specify exact positions for the points.

To edit the polygon, select it by any of its points. If you then hold the left mouse button and move the mouse, you will move the point. If you also hold <Alt>, you will move the entire polygon (and any other polygons/splines that are selected). See the tutorials for further explanation.

Limits: Drawing is limited to one plane at a time.

Name: FreeHand (Spline)
Access: LMB or <TAB> key when a spline is selected
Secondary: RMB (Manual Entry)
Keyboard: None

Brief: Draw freehand 3D splines.

Clicking this gadget allows you to draw splines. The new spline will be started at the current Attach Point. Pressing and holding the left mouse button will drag a new point out from the last one. Tapping the <TAB> key will leave this point at its current position and create a new one. This may be done any number of times to create a spline of any number of segments. The plane that the new point moves in is controlled by the Active Axis. If Y is selected, movement of the point will be in the XZ plane.

If you select an existing spline, then tap the <TAB> key, a new point will be created in the selected spline and you will enter FreeHand mode. If you want only one additional point, you can click the right mouse button on the Delete Point gadget. Additional points may be created by additional taps on the <TAB> key or additional right mouse button clicks on the Delete Point gadget. The point is not attached to the mouse pointer. The mouse pointer simply indicates a direction and amount of movement that you want for the point. If you select the FreeHand gadget with the right mouse button, you enter Manual Mode where you can specify exact positions for the points.

As you move a new point, the spline is created exactly the same as a polygon, but there are two red lines that run from each point to a third of the length of the side you are creating. These are the lines to the control points. After you have created the spline, you can move these control points to curve the spline.

To edit a spline, select the spline by any of its points or control points. Hold the left mouse button and drag the mouse to move the point you have selected. If you also hold the <Alt> key, the entire spline (and any other selected ~~polys~~polygons/splines) will move. If you hold down <Ctrl>, and if you have selected a point, the point and the nearest control points will move. Without the <Ctrl> key, if you have selected a control point, the opposing control point will be placed collinear with the control point you chose and its point, and at the same distance from the point, and will move accordingly, allowing you to smooth the spline. See the CSpline tutorial.

Limits: Drawing is limited to one plane at a time.

Name: Flat View
Access: LMB
Secondary: RMB (View Angle)
Keyboard: <Spacebar>

Brief: Convert to a Flat View

This gadget converts the view so that you are looking directly from the end of the currently selected Active Axis.

This is the proper view for drawing and moving. Normally, after you have changed the Active Axis gadgets, you would select the Flat View gadget to get the proper view to draw in or move objects.

As explained in the tutorials, you change view angle through the number pad or number keys. The Flat View gadget is a convenient way of hitting these special views easily and quickly. You do not have to be in flat view to draw or move, but it can be confusing if you are drawing in a plane you are not looking directly at in a Flat View. Use of the right mouse button allows you to manually enter the view angle.

Limits: None

Name: Center
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Center view, or selected polyspolygons
If you select this gadget with the left mouse button, it will center the view to the screen. If you select this gadget with the right mouse button, it will center any selected polyspolygons to the current Attach Point.

Limits: None

Name: PageMove
Access: LMB
Secondary: RMB
Keyboard: <Shift> <Cursor Up>, <Cursor Down>, <Cursor Right>, <Cursor Left>

Brief: Move the view. Secondary purpose to center Attach Point to selected polyspolygons (or origin if no polygons are selected).

Selecting this gadget allows you to move the view position left to right, up and down, similar to a page move in a flat drawing program. After selecting the gadget, move the pointer into the window and while pressing and holding the left mouse button, move the pointer. The view will move with it. A secondary use of the gadget is to center the Attach Point to any selected polyspolygons. For example if you select a sphere and then click the right mouse button on the PageMove gadget, the Attach Point will move to the center of the sphere. This is particularly useful for path placement, centering objects in conjunction with the Center gadget, etc.

If no polyspolygons are selected, clicking the right mouse button places the Attach Point at the origin.

Limits: None

Name: Zoom (2)
Access: LMB
Secondary: RMB
Keyboard: <Cursor Up> <Cursor Down>

Brief: Control magnification of the view.

These two gadgets are the primary way to control the magnification of the view. In actuality, they move the Editor's virtual camera closer to and farther from the origin. In effect, they increase and decrease the apparent magnification. A single press on the gadget with the "UP" arrow will halve the magnification. A single press on the "DOWN" arrow will double the magnification.

Using <Cursor Up> and <Cursor Down> keys (unshifted) will result in the same action but at 1/10th scale.

Clicking the right mouse button on either gadget will open the observer position requester allowing you to enter the magnification manually. "Image Distance" is the variable changed by these gadgets. This variable is the actual distance of the virtual camera from the origin. A typical value is 160000. The other variable controls the perspective of the view. You can increase the perspective by tapping the cursor left key and decrease it by tapping the cursor right key (without this requester open). The two variables are tied together.

Try altering the perspective with the cursor keys, then opening this requester to see the new values, closing it, changing the perspective again, and looking at the values again. You will see that both values are changed. In general, you should not use too much perspective in the Editor and rendered views, since increased perspective requires the virtual camera to be close to the origin increasing the possibility of polygons moving behind it. ~~Polys~~Polygons that move behind the virtual camera will not render properly - unless you add a camera.

If you use an added camera, the view you see in the preview and render is from the camera you add. The Editor view is always from the virtual camera. Use an added camera when you want to emphasize perspective in the render, or move the viewer position freely, with polygons behind the view position.

Limits: "Image Distance" must be positive. If the polygons move behind the virtual camera position, they are approximated. Use of a camera will prevent this during the render and anim preview modes, but not in the Editor. Normal operation of the program keeps the perspective to a minimum in the Editor and uses magnification to do detail work, instead of moving the virtual camera too close.

Name: AreaSelect
Access: LMB
Secondary: None
Keyboard: None

Brief: Specify an area of view

If you want to "zoom in" on a specific area of the drawing, use this tool. After clicking on the gadget, move the pointer to a point on the screen that represents the upper left corner of the area desired and click (don't hold) the left mouse button. Move the pointer to the lower right of the area desired. You will see a "rubber band" box that follows your mouse movement. When satisfied with the area defined, click the right mouse button. The view will change page position and magnification to represent the desired view area.

See the related topic, Track Attach Point under the Edit menu. It is a specialized tool for keeping an area in view regardless of its position in space.

Limits: The area desired must be on screen to draw the box around it. If not, zoom out until it is visible before using the tool.

Name: Multiple Select
Access: LMB
Secondary: RMB (Select All)
Keyboard: None

Brief: Select Polygons (ignore groups)

Multiple Select allows you to use a rubber band box around a group of polygons to select those within the box's boundaries. The polygons will be selected regardless of their group. This allows you to, for instance, set the color of part of a sphere to a different color than the rest of the sphere, without regrouping.

To use, select the gadget with the left mouse button. Move the pointer into the view window to a position representing the upper left of a bounding box surrounding the desired ~~polys~~polygons. Click (don't hold) the left mouse button. Move the pointer to the lower right corner of the box. You will see the rubber band box form. When you are satisfied with the position of the box and have all the desired polygons contained in it, click the right mouse button. All polygons within the box will be selected.

Any previously selected polygons will still remain selected. As a secondary function, clicking the right mouse button on the gadget will select all polygons in the current space.

Limits: May not select special purpose polygons.

Name: Grid Snap
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Snap point or poly(s) to nearest grid locations.

This gadget toggles on and off. If on (depressed) the program will limit any movement of points to the grid you set in the defaults. The point will snap to the nearest intersection in the grid lattice. The lattice is NOT visible, but remember when using the grid that it is 3D. You may inadvertently move a point or poly(s) forward or backward with no apparent change on screen, especially if you are in isometric mode.

Click the Grid gadget with the right mouse button, to open the Grid Defaults window. Here you set the size and center position of the grid you wish to use as well as deciding which axes will be forced to snap.

The Size gadgets set the distance between points in the grid lattice. The Center gadgets set the position of the center point in the lattice, allowing the lattice to be aligned with any point in space you choose. By aligning to the current Attach Point the center gadgets will be set to the position of the current Attach Point in your drawing. The Applied Axis gadgets allow you to turn ON/OFF the snap in the three axes. If depressed, the snap is ON in that axis.

The Absolute gadget, if ON forces points to snap to the next point in the lattice set by the Center and Size gadgets. If this gadget is OFF, the program automatically sets up a lattice when you select a point in the program. If you hold down <Shift> and select a new point, the lattice is updated to the position of the new point you have selected. This allows you to select a point or object and move it in even increments (set in the size gadgets) without it snapping to the world grid. It is just as though you set the center of the grid to the point you selected.

Limits: None.

Name: Snap
Access: LMB
Secondary: RMB
Keyboard: <F1>, Secondary: <F2>

Brief: Snap point or poly(s) to current Attach Point

It is often desirable for a point or a polygon to snap to a points in space. This flexible tool enables just that. To use it, set the Attach Point to the desired location. Then select the point on the poly that you want snapped to that position. Selecting the gadget with the left mouse button (or <F1>) will snap the selected point to the Attach Point. Selecting the gadget with the right mouse button (or <F2>) will snap the entire polygon to the Attach Point (at the location of the selected point). See also the Center gadget for a center snap.

Limits: None

Name: DeletePoly
Access: LMB
Secondary: None
Keyboard: None

Brief: Delete selected polygons/objects

The DeletePoly gadget does just that. Select the polygons that you wish to delete, then click the DeletePoly gadget with the left mouse button. You will be asked to verify the operation.

All necessary cleanup is performed for you. If the deleted poly is a path, any polygons assigned to it will be unassigned. If the deleted poly has shared ~~polys~~polygons for shading, the share will be unassigned. Attribute and Texture lists will not be deleted, but will remain for any possible future application.

Limits: None

Name: DeletePoint
Access: LMB
Secondary: RMB (add point and enter Freehand mode)
Keyboard: None

Brief: Delete or add a point

To delete or add a single point to an existing polygon, select the point that you want to delete and click the DeletePoint gadget with the left mouse button. You will not be asked to verify, and this action cannot be undone. The point you have selected will be immediately deleted, leaving the polygon otherwise undisturbed. Similarly, if you select a point, and then click the right mouse button on the DeletePoint gadget, a new point will be added just after the one you've selected. Where the new point is added depends on the order of the points of the polygon or spline.

The new point will exist at the same point in space as the original. It is created "selected" and can be immediately moved to a new location. Once you add a point, the program enters Freehand mode, so you

can then use the <TAB> key to continue adding points~~.~~ Clicking the right mouse button on the DeletePoint gadget does nothing if you are in Freehand mode~~.~~ You must use the <TAB> key to add more points after the first one~~.~~ Limits: You may not delete or add a point to a polygon that has shading or textures applied to it~~.~~ Free these before using this tool.

Name: Group/Ungroup
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Group/Ungroup selected polygons

Group operations let you group polygons so they can be treated, in a sense, as a single object~~.~~ To use, just select all polygons that you want in a new group and click the left mouse button on the Group gadget~~.~~ The selected polygons will "set" to let you know the operation has completed~~.~~ Then if you select any polygon in the group, all polygons in the group will be selected~~.~~ Up to 65,000 groups can be maintained in any one drawing file (in any of the Group Levels as discussed below)~~.~~ You can still select one or more of the polygons within the group, without selecting the whole group, by holding down both <Ctrl> and <Shift> while selecting polygons~~.~~ ~~Polys~~Polygons are selected in the order created.

Once selected, a "partial group" can be moved, rotated, etc., independent of the rest of the group~~.~~ It still remains part of the original group, however, and when one of its polygons is selected without using the qualifier keys, the entire group selects.

The second half of this gadget displays the currently selected Group Level~~.~~ Group Levels 1 through 5 are "free" groups which you can use for any purpose~~.~~ For instance, you may want the shade of a lamp and its base to be grouped separately in group 1, but grouped together in group 2.

Group Levels 6 and 7 are special purpose groups~~.~~ For example, instead of "6" for level 6 you will see "Sha." This stands for "Shading Group." When you select some ~~polys~~polygons and set their shading to Gouraud or Phong, the program automatically groups them for you~~.~~ If you then want to reshade them, increment the Group Level to Sha and select one of the ~~polys~~polygons~~.~~ All ~~polys~~polygons that were shaded at the same time will select~~.~~ You cannot use the left mouse button on the group gadget if you are in the Sha level.

Level 7 is another a special Group Level~~.~~ If you increment to Level 7 you will see "Sdw." This stands for "ShadoW." You can optimize the program's shadow routines using this Group Level~~.~~ Before rendering a view, if Shadows are on, the program will examine all polygons in the drawing and find those that have the same Sdw group level~~.~~ It will then build a bounding cube around them and use this in its shadow feeler routines~~.~~ The shadow feeler routines will check intersection with this bounding cube (six tests) before looking at the individual ~~polys~~polygons in the cube~~.~~ For a 144-polygon sphere, this is a savings of more than 130 intersection tests for every pixel in the render that has a polygon that receives shadows!

Needless to say, the shadow groups are fundamental to understand when rendering shadows~~.~~ To create a Shadow Group, select all the ~~polys~~polygons you want grouped into a single bounding cube for the intersection tests, increment to the Sdw Group Level, and click the left mouse button on the Group gadget~~.~~ You can have up to 65,000 shadow groups in a single drawing. If you are not going to render shadows, this group can be used as a regular group in the Editor, although this practice is discouraged.

If you click the Group gadget with the right mouse button, you remove any grouping that was in place for the selected polygons in the selected group level. (Level Sha is freed by freeing shading for the

polysspolygons).. If you delete grouping, when you select a poly, it is the only one that selects. Limits: Maximum of 65,000 groups in each level.

Name: ShowAll
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Reveal any hidden polygons

Clicking this gadget with the left mouse button will reveal all polygons that have been previously hidden with the Hide gadget.. Clicking it with the right mouse button will reverse the Hidden and Show buffers.. Those polygons that are hidden will be shown.. Those polygons that are shown will be hidden.

Limits: None

Name: Hide
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Hide selected (or unselected) polygons

As a drawing increases in complexity, it is often desirable to remove some of the polysspolygons from the display to make the view clearer.. This gadget does this.. Just select any polygons that you want to temporarily remove from view and then select the Hide gadget with the left mouse button.. The selected polysspolygons will be hidden.. In some cases, it may be easier to select those polygons that you want to remain visible.. If so, select them and click the right mouse button on the Hide gadget.. The selected polysspolygons will remain visible, and the unselected ones will be hidden.

Hidden polygons remain unchanged during most operations.. They cannot be readily selected without revealing them.. However, if you select a polygon that is in a group which has hidden members, those hidden members will be revealed.. This can be used to advantage during routine work.. Hidden polygons are also hidden from the rendering process, so if you want to see only one object, hide everything else.

Hidden lights are not used in renderings, and if you hide the camera, Aladdin 4D will render the Editor view.. In general, hidden polysspolygons that represent special functions, like the lights or camera, will not render, but some, like paths, still function as normal.

Each space has its own hidden/showing polysspolygons and this information is also saved in drawing files.. You can use the Editor menu item Information to see whether a drawing contains any hidden polysspolygons.

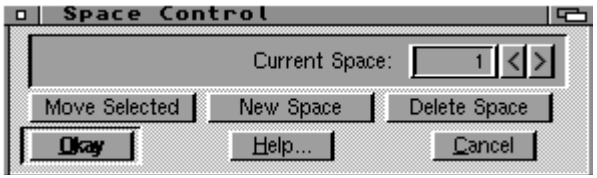
Limits: None

Name: Space Controls
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Create, delete and move through spaces

One of the most useful features of Aladdin 4D is its ability to maintain more than one drawing "space" (like layers in a CAD program).. This ability is like having more than one page in a drawing program, except that each "page" in Aladdin 4D is a complete 3D universe.. Each space can have its own lights, camera objects and paths.. Backgrounds, foregrounds and overlays, as well as Render Settings, and most lists (attribute, texture, CSpline, etc.) are common to all spaces.. Each space can be rendered separately, which can help in planning and story boarding an animation.

Polygons of any type can be "jumped" from one space to another.. This is most useful when working on complex objects.. As the number of polygons in the drawing increase, the feedback time also increases.. Using an additional space to work on a specific area of the drawing eliminates this problem.. If you are modeling an F-15 aircraft, for example, you can work on the wing separately, since while working on it, you'd have no need to see the fuselage and tail.. You can just create a new space and jump the wing to it, work on it there, and when done, jump it back to the original space.



The program starts with one space.. To create an additional one, click the right mouse button over either of the arrows to bring up a Space Control ~~requester which~~requester, which allows control of the spaces.. In this requester, you can create, delete, and re-order the spaces.. Click the "create" gadget to make a new space, for example.. If you have any polygons selected, the program will ask if you want to move them.. A new space is created, and its number appears in the middle space control gadget.. You can move through the available spaces by entering another number, or by clicking one of the arrows with the left mouse button.

If you delete a space, a requester asks for confirmation.. The space deleted is the CURRENT space - NOT the one displayed in the gadget.

For additional uses for spaces, see the section on saving .geo files.

Limits: Only one camera and target system is allowed per space.. All ~~polys~~polygons assigned to a path must be in same space.. ~~Polys~~Polygons in the same Shadow Group should be in the same space when rendering.

External Tools Gadgets

Aladdin 4D divides its tools into two groups. These are the built-in tools such as FreeHand, Flat View, Zoom, Page Move, Space Controls, etc., and the External tools. The external tools are accessed by the gadgets in the External Tools window gadgets in the External Tools window access the external tools. Its window is movable, resizeable, and closeable. If you close it, you can get it back with the appropriate option in the program's Settings pull-down menu.

Why are these tools called "external"? ? It's because they're contained in separate modules on disk, rather than in the main body of executable code. The common parlance for such tools - which will eloquently explain their reason for being external is is "plug-ins."

Aladdin 4D makes extensive use of the Amiga's multitasking system. These tools are actually separate programs that you run by clicking on the gadgets. When you click an external gadget, Aladdin 4D gives the tool access to its databases. The tool can then alter polyspolygons and objects that belong to Aladdin 4D. Actually Aladdin 4D is itself a tool, and the databases belong to a set of libraries that it maintains.. but let the programmers worry about all that!

Another nice feature is that external tools can be added or modified and distributed without updating the entire program.

Here we will document the standard external tools that come with the program. There may be others among the gadgets. If so, or if you receive a new tool through some other source like the Internet or a computer Bulletin Board Service (BBS), click the About gadget in the tool's default requester for instructions on its purpose and use. You may want to print out these third-party About files and add them to your manual.

You can also read the About files for all the external tools for any updated information changed since this manual was printed. Of course, this is especially true if you see a gadget that is not mentioned here.

There are three basic types of external tools.

1. Non Interactive: When you click the tool, the action occurs
2. Interactive: When you click the tool, you control the amount of the action by dragging the mouse. Polygons to be changed must be selected before clicking on the tool.
3. Interactive Selectable: Like interactive, but allows you to select polygons after clicking on the tool.

You can tell what kind of tool each is by looking at the tool's name, as printed in the gadget.

1. Just the name of the tool, no extension, means non-interactive.
2. Tool's name plus the extension .a (dot a) - an interactive one.
3. Tool's name plus the extension .as (dot as) - interactive selectable.

An example of a non interactive tool would is Clone. To use it, select the polyspolygons you want to replicate and then click the tool. Bingo! It's done. No interaction needed or requested.

An example of an interactive tool is Rotate. To use it, select the polyspolygons you want rotated, click the tool with the left mouse button and move the mouse pointer into the Editor window, press and hold the left mouse button and drag the mouse. Watch while the polyspolygons rotate. When they get to where you want them, let go of the button. When done with the tool, use the "set" command (right mouse button with the pointer in the Editor requester), or press <Esc> to undo the change.

Another example of an interactive selectable tool is the Slant tool. To use it, first click the tool, then select the polyspolygons you want to work on, then move the mouse pointer into the Editor requester, press and hold the left mouse button and drag the mouse to make the polyspolygons slant as you watch. When done use the "set" command (right mouse button with the pointer in the Editor requester), or press <Esc> to undo the change.

Usually, interactive selectable tools can be used on polyspolygons that have been selected before running the tool, just like non-selectable tools.

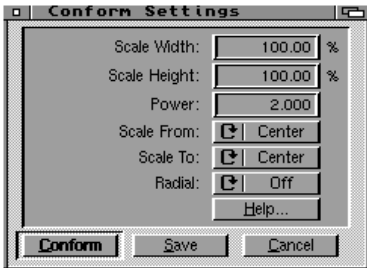
External tools gadgets can also be selected with the right mouse button. In such case, their behavior is generally different.

Aladdin 4D informs the tool, when it runs it, whether you have selected it with a left or right mouse button. If the tool adheres to the programmers' guidelines, it will operate normally when selected with a left mouse button, and it will open a default requester when selected with the right mouse button.

In this default requester, you can set defaults, read the About file, and usually accept the changes for the next use, cancel the changes, or even perform the operation on any currently selected polygons.

Remember, specifics for each tool differ, based on the purpose of the tool, and the current revision of the tool, so be sure to see the About file for each tool before you use it.

~~/// TEK: Duplicated in menu reference (p. 131)~~
Name: Conform (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Conform selected polyspolygons to a spline

This tool allows you to conform all selected polygons to a previously drawn spline Select the tool with the right mouse button to get the tool's default window. The gadgets in this window are:
documented on page 129.

~~Scale Width / Height These gadgets allow you to enter absolute scaling values. These values are only used in the non-interactive (perform) mode.~~

~~Power: This is function influences the "roundness" of the polyspolygons as the distance of the point increases from the plane of the center or Attach Point as specified.~~

~~Scale From: This allows you to apply the Power drop-off from the center of all selected polyspolygons or from the current Attach Point.~~

Scale To: In circumstances where the spline has two (or more) edges in line between the "from" point and the moving point, this allows you to choose which part of the spline to move toward. . If you choose FAR, the point will converge on the segment of the spline farthest from the point. . If you choose NEAR, the points will converge on the segment of the spline nearest the point.

Radial: This fundamentally changes the amount of movement of the points related to their distance from the FROM point. . If Radial is OFF, the points nearest the FROM point move the most. . If Radial is ON, these points move the least.

Conform looks at the spline you have chosen and decides whether it has been drawn in the XY, XZ, or YZ plane. . If it has been drawn in the XZ plane (usual in Y Flat View), the Conform tool will then converge the points of the polyspolygons outward toward the spline shape in the XZ plane that the point is in when you start using the tool. . The same reasoning applies if the spline is drawn in the other two axes.

Normal Use: To use the tool, draw a spline you want conformance to, just as though you are drawing a cross section of a shape. . Place the spline around the polyspolygons you wish to have conform to the shape. . Select the SPLINE first, then select any polyspolygons you wish to converge. . You are ready to use the Conform tool. . You can use it interactively, or in non-interactively.

To use interactively, click the Conform gadget with the left mouse button. . You'll see a window open that will show a conform buffer being built. . When the window closes, move the mouse pointer into the Editor window, hold down the left mouse button, and drag the pointer. . The scaling that occurs is like that of the Scale tool, but it is modified by the spline. . If you hold down <Shift> while you drag, the amount of scale is applied equally to both pertinent axes. . Without <Shift>, scaling is applied to both axes independently, based on mouse movement.

To use the non-interactive "perform" mode, click the Conform gadget with the right mouse button. . In the Conform defaults window, set the values you want and click Perform. . You will see the buffer being built, then the Conform will occur. . If you don't like the results, hit <Esc> before setting the polyspolygons. . If you like the results, use the Set command (RMB with mouse pointer in the Editor window) to make the changes permanent.

This tool changes non-triangles to be severely non-planar. . After using the conform tool you should almost always change the polyspolygons you have conformed into triangles before rendering, to avoid non-planar rendering artifacts.

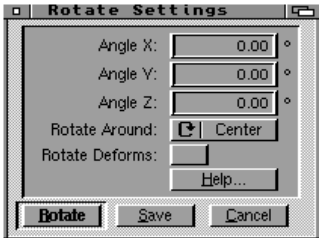
Limits: Only works with polygons. . Some special polyspolygons are not affected.

Tip: The Conform tool converges the points of the selected polyspolygons as though they are moving outward toward a cylinder whose cross section is shaped like the spline. . This means that if you anticipate the direction of the points in your original object, you can achieve a much more natural appearance in the finished conformed object.

To see this, check the examples demonstrated earlier in the manual.

/// TEK: Duplicated in menu reference (p. 125)

Name: Rotate.a (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Rotate selected ~~polys~~polygons

This tool allows you to rotate all selected polygons. When you select the tool with the right mouse button, you get the rotate tool default requester~~. .~~ ~~The gadgets~~ Page 122~~in this requester are:~~ documents this.

~~ROTATE XYZ: Here you can choose the number of degrees of rotation around each axis. .~~
~~Polys~~Polygons ~~must already be selected and you must use the Perform gadget for this to take effect.~~

~~ROTATE FROM: toggles from Center to Attach Point. . If Center, the program rotates all selected polys~~polygons ~~around their common geometric center. . If Attach Point the program will rotate all selected polys~~polygons ~~around the current Attach Point.~~

~~Rotate Deforms: here you can choose whether you want Rotate to affect any deforms that may exist for the selected polys~~polygons ~~at the same time you rotate the polys~~polygons~~. . THIS IS NOT UNDOABLE.~~

~~PERFORM: Close the requester and perform set changes, if any.~~

~~ABOUT: open the About file reader and read the information file for this tool.~~

~~CANCEL: close the requester with no changes~~

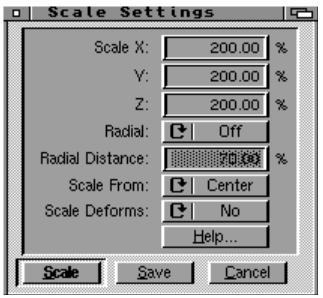
~~ACCEPT: close the requester with the changes made. . The changes will take effect next time the tool is used.~~

You can also use this tool in an interactive mode.

To do this, if the default is to rotate around the Attach Point, select and set some point on one of the polygons~~. .~~ This is now the current Attach Point~~. .~~ If the default is to rotate around the center, this is not necessary~~. .~~ Select some polygons~~. .~~ Click the gadget with the left mouse button~~. .~~ Move the pointer into the view window and press and hold the left mouse button~~. .~~ Drag the mouse and you will see the polygons rotate~~. .~~ The rotation is in the plane described by the currently selected Active Axis~~. .~~ If the Active Axis is Y, the rotation occurs in the XZ plane~~. .~~ Rotation occurs in XYZ order.

Limits: None

~~**/// TEK: Duplicated in menu reference (p.125)**~~
Name: Scale.a (or resize) (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Resize selected polygons

This gadget allows you to resize any selected polygons along any axis or combination of axes.

You may use this tool interactively. To do so, select the polygons you wish to change, then click the tool gadget with the left mouse button. Bring the pointer into the Editor window, press the left mouse button, and drag. You will see the polygons changing size as you drag.

If you use the tool with the Y axis selected, you will be able to change the Z dimension by dragging up and down, and the X dimension by dragging left and right. Similar changes occur with other axes selected. By holding down the <Alt> key, you change BOTH axes dimensions by the same amounts at the same time. By holding down <Shift>, you change all three axes dimensions by the same amount at the same time, regardless of the Active Axis.

If you click the gadget with the right mouse button, you will open a window that allows you to enter exact resize amounts in all three dimensions. To use this, select your polygons first, open the window and enter the desired percentages, then perform. You will see the polygons change.

The gadgets in the window are documented on page 122.

SCALE XYZ: the desired size. 100.0 is 100 percent, or no change. Be careful here, if you resize very small, so that adjacent points on an object become the same, you cannot scale them back up again to reclaim the original object. This is, however, common practice for one level of a deform when you want an object to vanish in size during an animation.

Radial: If this is engaged, the program will measure all selected polygons and describe a virtual sphere that will just enclose them. During the resize, the points of the polygons will be driven toward (or away from) the nearest point of the sphere. This mode can take advantage of "extender" polyspolygons**. Also see the extenders and how they are used in the Stretch tool illustrations.**

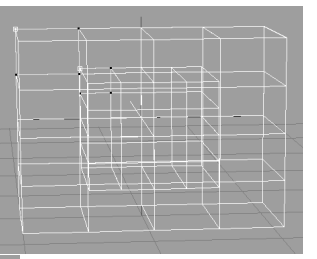
SCALE FROM: you can toggle this from Center to attach Point. If Center, the program scales all selected polyspolygons** around their common geometric center. If Attach Point the program will scale all selected polys**polygons** around the current Attach Point.**

SCALE DEFORMS: If this is ON, any polyspolygons** that are selected and have deform levels will have their deforms scaled at the same time as the polys**polygons**. If OFF, the deforms will remain unaffected. This is NOT undoable.**

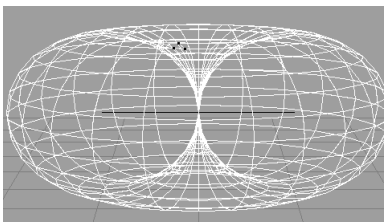
Limits: None

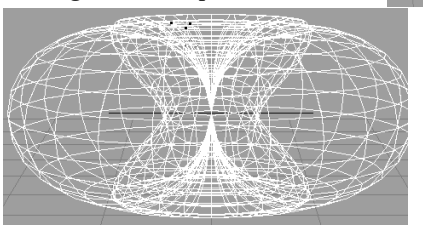
Some examples of the Scale tool:

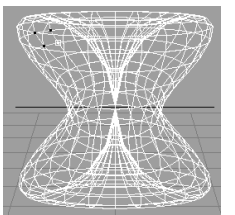
A simple cube, and the same cube resized freehand along two dimensions with the radial option OFF.



These illustrations show the resize tool in operation using the radial option. The original gridded cube was resized to 50 percent on all three axes forcing the points inward based on their relative distance from the virtual sphere. As you can see, the corner points, which would actually be on the surface of the virtual sphere are basically unchanged in their position, but







those points on the face which would be farthest from the sphere have moved the most~~.~~. The second illustration is the same gridded cube resized to 200 percent with the radial option on.

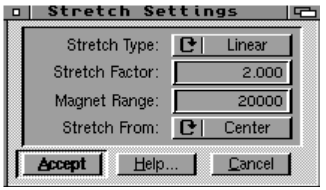
The simple torus on the left was resized using the radial option in a freehand manner~~.~~. The points around the outer edge of the torus are closest to the virtual sphere, so they move the least~~.~~. The result is that you can actually close the hole in a torus, if you want to.

~~/// TEK: Duplicated in menu reference (p.126)~~
~~**Shear.a**~~
Name: Shear.a (or slant) (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: See documentation on page 123.

~~/// TEK: Duplicated in menu reference (p. 127)~~
Name: Stretch.a (non-interactive or interactive)

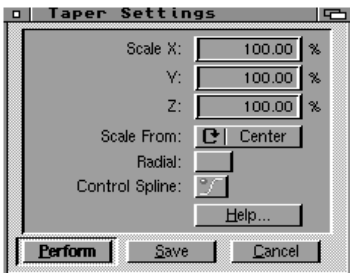
Access: LMB
Secondary: RMB
Keyboard: None



Brief: See documentation on page 124.

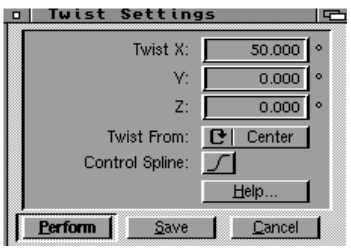
~~/// TEK: Duplicated in menu reference (p. 127)~~
Name: Taper.a (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: See documentation on page 125.



~~/// TEK: Duplicated in menu reference (p. 128)~~
Name: Twist.a (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

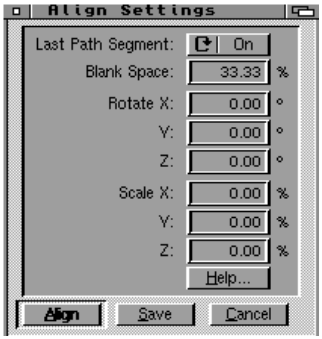
Brief: See documentation on page 125.



~~/// TEK: Duplicated in menu reference (p. 138)~~
Name: Wirebend.a (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: See documentation on page 135.

~~/// TEK: Duplicated in menu reference (p. 195)~~
Name: Align (non-interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: ~~/// ??? 0510 /j~~Align selected polygons along a path.

This tool can be used with either a LEFT MOUSE BUTTON which uses the stored defaults or with the RIGHT MOUSE BUTTON to change the defaults before use.

The tool is designed to think of positive Z as "UP". You should work with your extruded letters facing you from a flat Y view.

The path should be drawn from the flat Z view. The path does not have to be flat, the letters will align with vertical deviations.

The align will center to a point half way along the length of the path as described under BLANKSPACE below.

If you want an arc that reads from the front, still use this orientation, but use a Rotate X of 90° (or neg 90°), then use the normal rotate tool to rotate the arc into position.

NOTE that if you perform the align and the text reads the opposite way of that you desire, use the escape key to undo it, then use the TOOLS menu item "Reverse Points" on the path. When you perform the align again, it will read the way you want.

NOTE that the spacing between the object groups along the align path is relative to that along the X axis of the groups before the align. Vertical spacing is kept constant only for the individual groups, so if you have several lines of text, you must align them in separate aligns, or they will converge vertically.

Preparing Polygons:

As you know, "objects" in Aladdin 4D are composed of single polygons. If the align tool were to align single polygons, the extruded letters would be reduced to random appearing single polygons along the align.

You need to let the program know what you consider to be single objects.

You do this by grouping.

When the tool runs, it resizes ALL THE SELECTED POLYGONS IN EACH GROUP it finds around the group's center, and rotates and translates the group by its center.

The tool looks in the currently active group level.

This means you do not have to have the extruded letters align, but can also align complete words, or even phrases if desired.

To group the letters into separate groups, you must select all the polygons that compose a single letter, and group them.

To help you accomplish this, the extrude tool has an option called SINGLEGROUPS. If on, each selected poly, when extruded will be given a separate group, along with its extruded shell and cloned face. NOTE that this is done in GROUP LEVEL 3 ONLY AND ONLY IF YOU HAVE SINGLEGROUPS ON.

NOTE that you may still have to select and group a few, example;~~for instance,~~ the "stem" and the "dot" of the letter i.

Last Path Segment: If ON, the last segment of the path poly will ~~participate~~include in the align. If OFF, it will not.

Blank Space: When you select the path poly to align to, the program will measure its length, then subtract the percentage of this length you specify here. The polygons will then be resized to fit the remainder of the path length, and moved into position along this remainder.

NOTE that the polygons are ALWAYS CENTERED to the center of the path as measured from the path's firstpoint to its lastpoint if the last path seg is OFF, or back to the firstpoint if the last path seg is ON.

So if you, for example, have a circle flat in the XY plane, with its first point at the top of the screen when viewed in the Z flat view, and have the blankspace set to 0.5 (50 %) the extruded letters will be fitted to the bottom half of the circle as viewed from the Z flat view.

Rotate X,Y,Z: Specify any angles of rotation you want.

NOTE that the tool is designed to work with the positive Z axis "up". This makes it easy to generate logos such as that used in the UNIVERSAL pictures movie header. You may also generate arcs that go overhead and read properly from the front by specifying an X rotation of 90°, or negative 90° depending on the side you want to view it from. After performing the align, you then rotate the aligned letters as a unit around the X 90° (or -90°) using the standard rotate tool.

Partial rotations are also quite useful for special positioning.

Scale X,Y,Z: Specify any scaling values you want. Note that 1.0 is unity, or no change. 2.0 would be twice, and 0.5 would be half the original size.

Note that the polygon "objects" will automatically be scaled to fit the length of the path after the blankspace has been subtracted. You may, however, want to reduce them even more, or enlarge them, in any of the three axes. This can adjust the "letterspacing", or cause them to fit the smallest curve in the path better, or even cause them to overlap for some interesting effects.

/// ~~Duplicated in menu reference (p. 120)~~

Name: Breakup (non-interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Automatically "face" selected ~~polys~~polygons with triangles - that is, simplify complex polygons.

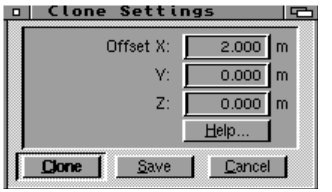
This tool allows you to convert all selected polygons into triangles~~.~~. This is important if you are going to deform a texture-mapped complex poly (more than four points), or if you have non-flat ~~polys~~polygons, or if you want faster render speed than would be normal with ~~polys~~polygons of more than 4 points.

Select the tool with the right mouse button to get the Breakup Settings requester - but there are no defaults for this tool~~.~~. The window opens so you'll have access to the About file for this tool.

Limits: ~~Polys~~Polygons that are too degenerate, i.e., very non-planar, or looping back on themselves, will create errors in the faces generated, or may be incomplete.

/// TEK: ~~Duplicated in menu reference (p. 120)~~

Name: Clone (non-interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Make an exact copy of selected polygons

The Clone tool is the same as the Edit / Tools menu item Clone~~.~~. Use the right mouse button to bring up its settings requester, or the left button to operate it on the selected polygons.

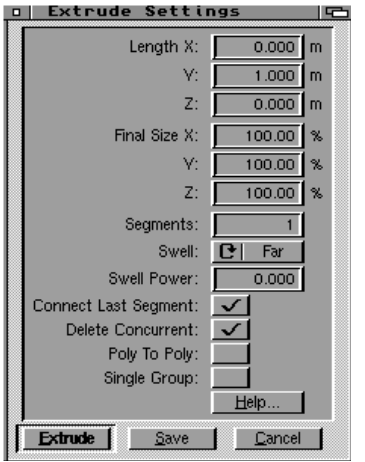
If the offsets for the clone are at 0, 0, 0 you may not see the new ~~polys~~polygons, as they will be at the exact position in space as the originals.

The gadgets in the Clone Settings requester are:

Offset X, Y, and Z: Specify positional offsets for the newly created polygons.

It is permitted to select only portions of a group for cloning~~.~~. All newly cloned ~~polys~~polygons will be assigned new groups in a logical manner.

Limits: Only one camera is permitted per space~~.~~. You may not clone a camera or target~~.~~. Create additional targets using the camera control.



/// TEK: ~~Duplicated in menu reference (p. 130)~~

Name: Extrude (non-interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Extrude selected polygons.

If you click this gadget with a left mouse button any selected polygon(s) will be extruded according to the current extrude settings. If you click this gadget with the right mouse button, the extrude settings requester opens. Here you can edit these defaults to your purposes. The gadgets in the requester are documented on page 119.

Length X, Y, and Z: Controls the distance to extrude along each axis.

Final Size X, Y, and Z: Final Size controls any desired reduction or enlargement as the extrude occurs.

Segments: The number of segments created can be specified and should be non-zero.

Swell: The end of the extrusion you want to swell. You must enter something in the Swell Power gadget for this to work.

Swell Power: Causes the extrusion to occur according to a power function - for some useful shapes.

Connect Last Segment: The Connect Last Segment switch tells the extruder whether the segment of the selected polygon between the first and last points should be extruded. For instance, if you extrude a square with this ON, you will get a square pipe. If it is off, you will get a pipe with one side missing, like a rain gutter.

Delete Concurrent: If ON, after the extrude is performed, all new polyspolygons will be examined and if any are in exactly the same space, they will be deemed to be interior to the extrude and will be removed. This is very important for fonts, which often have lines bridging the outside of the letter with the inside, like the letter "O". If you extrude this letter with Delete Concurrent OFF, you will get two extra polyspolygons at this "seam", and they will play havoc with the shading routines. In general use, leave this gadget ON.

Poly To Poly: If this is ON, the program will look through all the polyspolygons in the drawing and determine whether you have exactly two polyspolygons selected. If this is true, it will count the points of both polyspolygons to determine whether they are the same. If this also is true, it will create a shell between the two polyspolygons, on a point for point basis, starting with their first points. This allows you to create "cross sections" of objects, and then "skin" them.

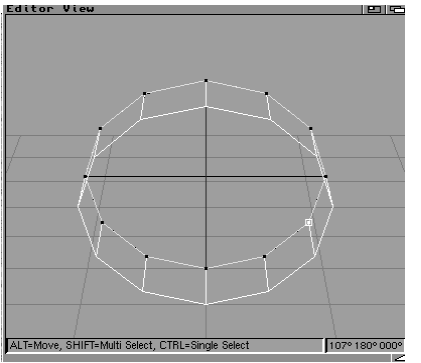
Single Group: Makes the extruded object a single group.

If you don't like the extrude produced use "DeletePoly" before setting the changes, to retain the original shape.

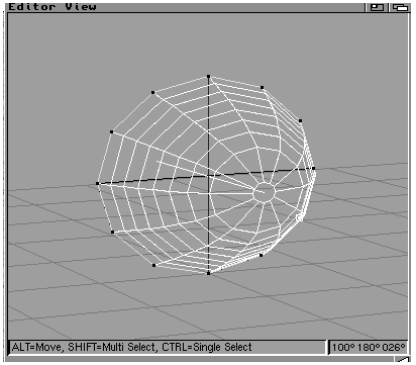
Limits: If the extruded polyspolygons are assigned to paths, the new polyspolygons will automatically be assigned to the same path, unless the path uses deforms.

Some typical examples of extrudes:

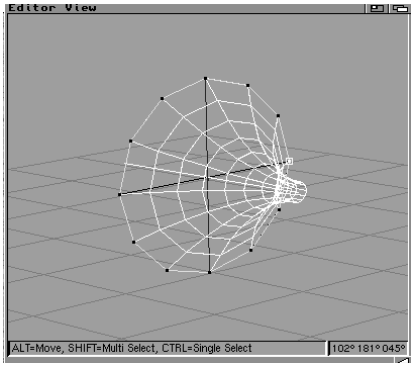
Offsets: 0 10000 0
Final Size: 100 100 100
Segments: 1
Connect Last Segment: ON
Swell: Near
Power: 0.0
(Attach Point at top point)



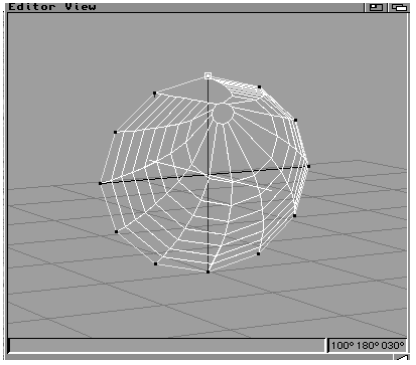
Offsets: 0 10000 0
Final Size: 10 10 10
Segments: 8
Connect Last Segment: ON
Swell: Near
Power: 2.0
(Attach Point at center of poly)



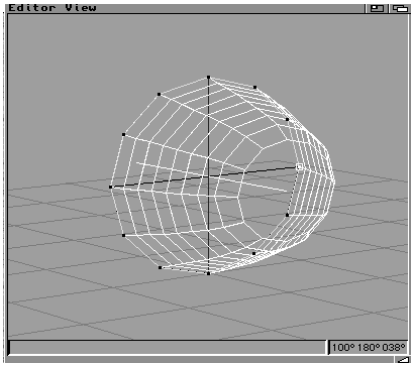
Offsets: 0 10000 0
Final Size: 10 10 10
Segments: 8
Connect Last Segment: ON
Swell: Far
Power: 2.0
(Attach Point at center of poly)



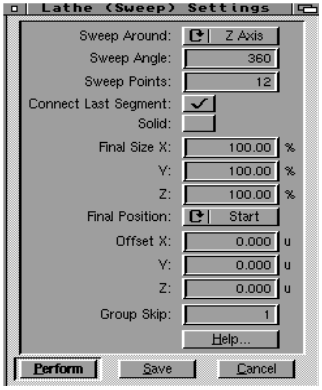
Offsets: 0 10000 0
Final Size: 10 10 10
Segments: 8
Connect Last Segment: OFF
Swell: Near
Power: 2.0
(Attach Point at top point)



Offsets: 0 10000 0
Final Size: 50 50 50
Segments: 8
Connect Last Segment: ON
Swell: Near
Power: 1.5
(Attach Point at center of poly)



~~/// TEK: Duplicated in menu reference (p. 122)~~
Name: Lathe (aka Sweep) **(non-interactive)**
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Create a dimensional version of a polygon by spinning (sweeping) it around a center - a rotational extrude of the selected polygons.

One of the most useful tools in 3D modeling is the rotational extrude called "Lathe." This allows you to create all types of "round" forms from a single polygon by rotating the polygon around a selected point in space and creating an extruded shell along the points of the polygon as it rotates.

When you select this tool with the right mouse button, you open the Lathe's settings requester. The gadgets in this window are documented on page 120.

Sweep Around: Pick an axis, X, Y, or Z.

Sweep Angle: This is the size of the arc the Lathe will use in degrees. It can be negative if needed. If you use offsets, it can be greater than 360.

Sweep Points: This is the number of divisions the Lathe will produce.

Connect Last Segment: whether the last side of the poly being swept will create polyspolygons in the shell. Connect Last Segment tells the program whether to Lathe the last edge of the poly (between the first and last points).

This is of basic importance to the Lathe operation. If you make an arc of 180 degrees and spin it around one of its end points, you will get a sphere. But if you do the Lathe operation with Connect Last Segment ON, you'll get a "stem" running through the center of the sphere.

This stem will cause oddities during shading. For a closed form, like a sphere, you should turn Connect Last Segment OFF. If, however, you are Lathing a shape into an open form, like a torus (donut), and have Connect Last Segment OFF, you will produce a donut with an opening running all around it. For an open form, like a torus, you generally want Connect Last Segment ON.

Final Size X, Y, and Z: These allow you to do an enlargement or reduction of the original shape during Lathe operation.

Final Position: Choose from Start, Center, Outside, or Inside. These are only useful if you are using Offsets.

Offset X, Y, and Z: Offset allows you to do a translation of the shape as the Lathe operation occurs.

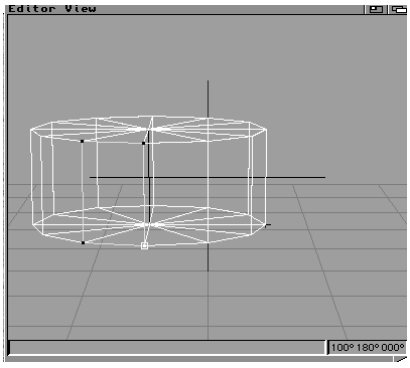
Group Skip: If set to other than 1, the polyspolygons are grouped in alternating manner based on this number.

The shape that you use the Lathe on is NOT deleted, and generally should be selected and deleted, or moved out of the form, after the Lathe operation is done.

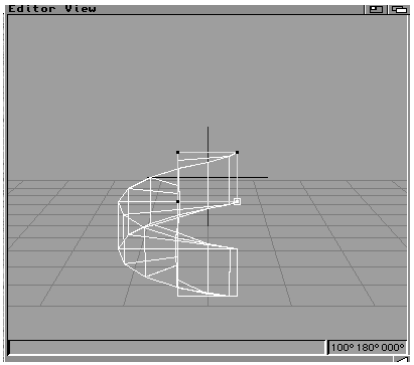
Limits: Lathe operations may occur only around main axes.

Some typical examples of Lathe operations and the shapes that made them:

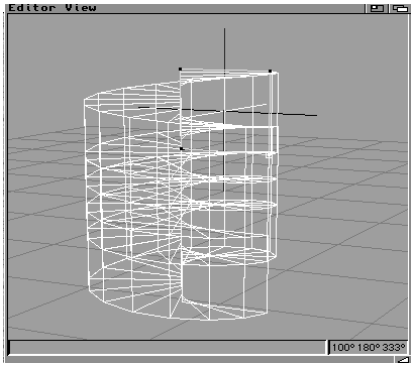
Sweep Angle: 360
Sweep Points: 12
Connect Last Segment: OFF
Solid: ON
Final Size: 100 100 100
Offset: 0 0 0



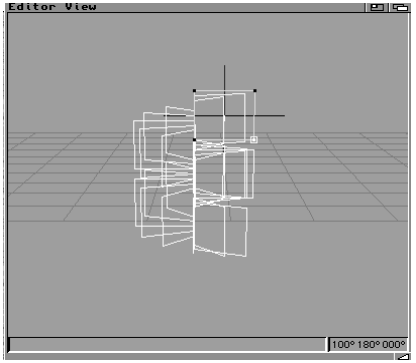
Sweep Angle: 360
Sweep Points: 12
Connect Last Segment: ON
Solid: ON
Final Size: 100 100 100
Offset: 0 0 0



Sweep Angle: 1080
Sweep Points: 48
Connect Last Segment: ON
Solid: ON
Final Size: 100 100 100
Offset: 0 0 20000

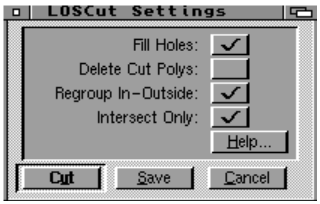


Sweep Angle: 720
Sweep Points: 24
Connect Last Segment: ON
Solid: OFF
Final Size: 100 100 100
Offset: 0 0 24000



The Lathe tool is the same as the Edit / Tools menu item Lathe~~.~~. Use the right mouse button to bring up its settings requester, or the left button to operate it on the selected polygons.

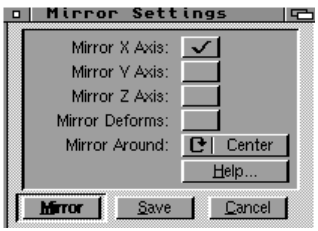
Name: LOScutter (Line Of Sight Cutter)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Cut a polygon with another shape.

The LOScutter tool is the same as the Edit / Advanced Tools menu item LOScut~~.~~. Use the right mouse button to bring up its settings requester, or the left button to operate it on the selected polygons~~.~~. See the Edit section for further details.

Name: Mirror
Access: LMB
Secondary: RMB
Keyboard: None

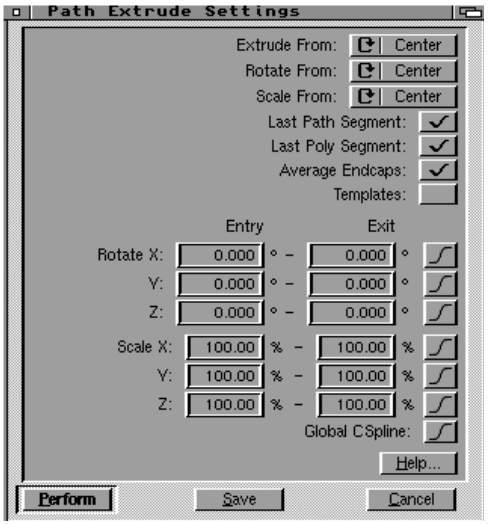


Brief: Create a mirrored copy of an object, or flip it.

The Mirror tool is the same as the Edit / Tools menu item Mirror~~.~~. Use the right mouse button to bring up its settings requester, or the left button to operate it on the selected polygons~~.~~. See the Edit section for further details.

~~/// TEK: See menu reference?~~

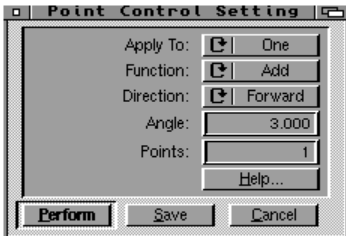
Name: Path Extrude
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Extrude a path to create polygons around the path.

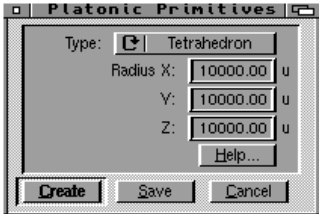
The various settings are all documented on page 193.

~~/// TEK: Duplicated in menu reference (p. 136)~~
Name: Point Control, formerly "PntCntrl"
(non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None



Brief: Reduce or increase the number of points in selected ~~polys~~polygons.

This tool allows you to increase or reduce the number of points in all selected polygons Select the tool with the right mouse button to get its default window~~.~~. The gadgets in this window are shown on page 133.~~:~~



~~Apply To: Choose whether the tool works on one poly at a time, or on all selected polys~~polygons~~.~~

~~Function: Choose whether to add or remove points.~~

~~Direction: if Apply To is set to 1, you are asked to choose a second point in the selected poly. . The operation is then applied between the initial point you selected the poly with, and the second point you choose. . If Direction is set to Forward, the program operates on the segments of the poly between the first and second point in order. . If it is set to Backward, the program works on the segments in reverse order.~~

~~Angle: (only when removing points) This allows you to choose a minimum angle (in degrees) that must be encountered between segments before the point between the segments is removed.~~

~~Points: (only when adding points) This is the actual number of points that will be added in each existing segment.~~

~~To use the tool in ONE mode, set the defaults to ONE. . Make sure you know which way the points are ordered, and whether you want the tool to work Forward or Reverse. . Choose the poly you wish to change by the first point in the area you wish to change, then click the tool with the left mouse button. . You will be prompted to choose the second point. . Do this, and you will see the changes you requested occur. . If you choose the same point, the entire polygon is done. . This action is NOT undoable, so you may want to clone the poly and hide it or jump it to a new space, so you can get it back if you need it.~~

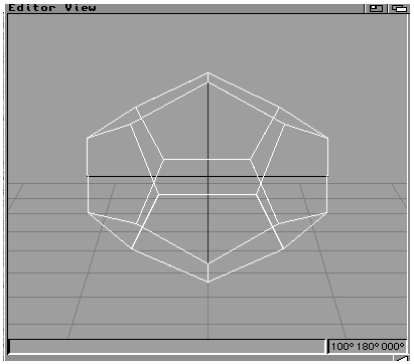
~~To use it in the ALL mode, set the defaults to ALL, select the poly(s) you want to change, and select the tool with the left mouse button. . You will see the changes occur.~~

~~Of course, if you have the poly(s) selected before changing the defaults, you can simply click the Perform gadget (you will be prompted if in the ONE mode), and see the changes.~~

~~Limits: Will not operate on most special polys~~polygons~~.~~

~~/// TEK: Duplicated in menu reference (p. 117)~~
Name: Primitive-Plat~~p~~ "Platonic Primitives" (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Generate objects from the Platonic primitives.



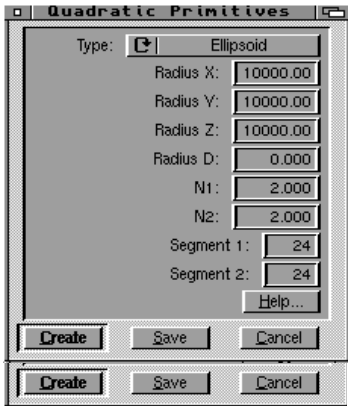
This tool allows you to generate Platonic polygons Select the tool with the right mouse button to get the tool's settings requester~~. .~~ The gadgets in this window are:

Radius X, Y, and Z: Choose exact sizes for the Primitive in all three axes~~. .~~ Some primitives may require conversion to triangles, especially if these are not equal.

Type: Clicking this gadget rotates through all primitives available.

If you have already set the defaults to those you wish, just select the tool with the left mouse button and the primitive will appear~~. .~~ You can also select the tool with the right mouse button to open the settings requester to choose options, then select the Perform gadget and the primitive will be generated~~. .~~ A Dodecahedron is illustrated.

~~Limits: None~~



~~/// TEK: Duplicated in menu reference (p. 117)~~
Name: ~~Prim~~ q~~Primitive~~-Quad "Quadratic ~~Primitives~~" (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Generate quadratic primitives

This tool allows you to generate any of several quadratic primitives~~. .~~ Click the tool with the right mouse button to get the tool's settings window~~. .~~ The gadgets ~~in this window~~ are:

Radius X, Y, and Z: Choose exact sizes for the generated primitive in all three axes~~. .~~ You may want to convert the generated primitive to all triangles before rendering, especially if these are not planar.

Radius D: This is used when the primitive has a "hole" in its center, like a torus~~. .~~ It is a percentage of the main radii~~. .~~ For a torus, a normal range would be 0.0 to 5.0 or so~~. .~~ Must be positive.

N1, N2: These control the "roundness" of two aspects of the quadratic~~.~~. A value of 2.0 gives a circular arrangement, 1.0 gives a linear appearance, and 0.5 gives a "coved" arrangement~~.~~. It is best not to use values below 0.5, or duplicate ~~polys~~polygons can be created as well as ~~polys~~polygons with multiple collinear points - which can cause rendering artifacting~~.~~. Values higher than 2.0 give a "rounded corner" appearance ~~and are quite useful.~~

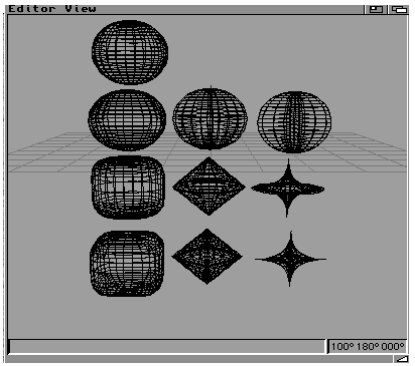
Segment 1, Segment 2: Control the number of polygons generated~~.~~. Segment 1 controls divisions around the rotational axis (like in a Lathe), and Segment 2 controls the divisions around the opposing axis.

Type: clicking this gadget rotates the quadratic types available.

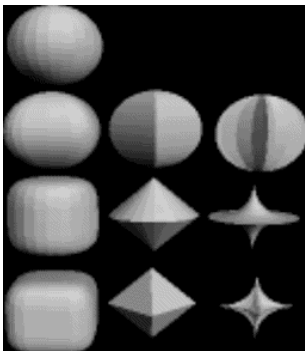
The generated primitives are grouped in a natural way for your convenience in shading~~.~~. In Group 1, they are grouped as a whole~~.~~. In Group 2, they ~~are~~ group~~ed~~ in quadrants, other levels have different arrangements.

If you have already established the settings you want, just select the tool with the left mouse button, and the primitive will appear.

Limits: Be careful not to set N1/N2 too low for the size object you are creating.



Here are some examples ~~of the output of this tool~~ showing the results of different values for N1, N2, and D.



Shown is an ellipsoid.

The N1/N2 ratio is:

top: 2/2
row 1: _____2/4, 2/1, 2/0.5
row 2: _____4/2, 1/2, 0.5/2
row 3: _4/4, 1/1, 0.5/0.5

N1 and N2 also affect all other quadratics generated~~.~~. Try rendering a torus with an N1/N2 of 4/4, and a D of 4~~.~~. Radius XYZ are each 5000~~.~~. Segment 1 is 48, Segment 2 is 24.

Name: ReadDrawNote
Access: LMB
Secondary: RMB
Keyboard: None

Brief: ReadDrawNotes created by the Extern handler DrawNote.

DrawNotes are comments and other useful information you can attach to a drawing file~~.~~. The DrawNote Extern Handler lets you include any number of variable-length text files in a drawing, to remind yourself about what you did, or what you meant to do, settings for certain attributes, render options, etc~~.~~. These notes are saved and loaded with the drawing~~.~~. You can use your choice of editor and reader programs to view the notes.

This is a special tool that launches the DrawNote handler in "read only" mode~~.~~. Just select a DrawNote and click the tool with the left mouse button~~.~~. It bypasses the DrawNote window and goes directly to the reader~~.~~. Use this when you only want to read a note, not edit it~~.~~. If you click the right mouse button on the tool, you get its About file directly~~.~~. There is no defaults requester for the DrawNoteRD tool.

Name: RunDpaint /
Name: RunImageFX
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Examples of multitasking Aladdin 4D with Dpaint or ImageFX two outside image processing (or paint) programs. We will focus on running ImageFX below.

This tool is another example of a script file you can write to launch your programs from the toolbox of Aladdin 4D. You do not have to be a programmer to do this. If you have ImageFX, it's also a very useful tool for launching it - that is, if you have enough memory in your Amiga to have both programs running.

The tool lives in a directory known to the Amiga as Aladdin 4D:tools. To look at it, open an Amiga Shell window. This gives you a prompt that looks like a greater-than (>) symbol. Issue the following command:

cd Aladdin 4D:tools

Be sure to type that exactly, including the one space, If you get an error message, either you have typed something incorrectly (caps and lowercase don't matter), or you do not have Aladdin 4D installed correctly. If you get a prompt, however, all is well and your Shell's "current directory" is where Aladdin 4D stores its tools. Note that this is NOT where you store ImageFX.

The next command is:

ed RunImageFX

The filename "RunImageFX" is the script name ~~of the script~~ that Aladdin 4D uses ~~to implement~~for the external tool "ImageFX."

The "ed" command loads the Amiga's simplest (and it is ~~*REALLY*~~ *really* simple) text editor. The ed window shows the content of the file:

run >NIL: ImageFX:ImageFX

This is an Amiga command that will launch ImageFX. The little script expects to find it on a hard drive named "work" and in a drawer on that hard drive named "ImageFX". This is the default installation for ImageFX. Again, caps and lowercase letters don't matter.

If you keep ImageFX on a hard drive named something other than "Work", edit this command line to fit your configuration. To save the file from ed, you can use the pull-down menus, or hit <Esc> <X>.

The next step is to set the little file's "protection bits" so AmigaDos will know it's a script. You can do this with most directory utilities, or use the Shell. The command is:

protect Aladdin 4D:tools/RunImageFX +S

In this command, the file's name is preceded by its complete path, so there's no confusion about which file is to be changed. The +S parameter tells AmigaDos to make that file usable as a script. After you set the +S protection bit on a text file, it can be executed from a Shell command line as though it were a program, its contents being considered to be AmigaDOS shell commands. Watch out, though. The slash in that command has to go the right direction - the right-HAND direction. No spaces, except the first one, caps and lowercase don't matter. Again, if you get an error message, you probably typed something wrong.

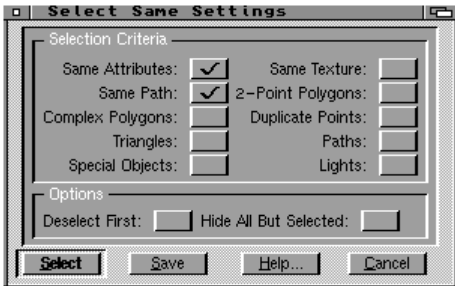
Why would you want to run another application from within Aladdin 4D~~?~~ ? If it's an application you run often, this gives you a quick way of getting it to open. You can then use it and close it, without going to the Amiga Workbench screen. ImageFX is quite useful for making quick textures, or editing them to

match the effect you're trying to achieve with the renderer, or adjusting the scaling, brightness, color balance, sharpness, etc~~.~~. of an image that is to be used as a bitmap texture.

You can use this method to add other tools to Aladdin 4D~~.~~. Just copy one of the existing tools by a new name (like "RunAnimViewer"), and then ~~use Shell to~~ ed~~(it)~~it the new file, substituting the path and filename you need.

If you don't want to use this tool, you can simply delete it from the Aladdin 4D:tools directory~~.~~. Then it will not show in the tool box.

Limits: You must have enough memory (graphics and fast memory) to run both Aladdin 4D and any other application(s) at the same time.



~~/// TEK: Dunno if this is in the menu reference or not.~~
Name: SelectSame (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Select ~~polys~~polygons based on same variables

This tool allows you to select polygons that share the same Texture List, Attribute List, etc~~.~~. Select the tool with the right mouse button to get the SelectSame Settings requester~~.~~. The gadgets in this window are:

Selection Criteria

Same Attributes: Will select all ~~polys~~polygons in the current space that use the same Attribute List as the currently selected poly~~.~~. Requires a single selected poly.

Same Texture: Will select all ~~polys~~polygons in the current space that use the same Texture List as the currently selected poly~~.~~. Requires a single selected poly.

Same Path: Will select all ~~polys~~polygons in the current space that are assigned to the same path as the currently selected poly~~.~~. Requires a single selected poly.
2-point Polygons: Will select all ~~polys~~polygons in the current space that have only two points.

Complex Polygons: Will select all ~~polys~~polygons in the current space that have more than four points.

Duplicate Points: Select all ~~polys~~polygons in the current space that have duplicate points.

Triangles: Selects all triangular ~~polys~~polygons

Paths: Selects all ~~polys~~polygons in the current space

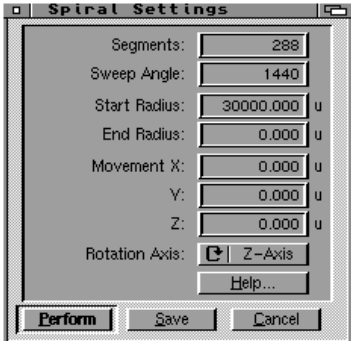
Special Objects: Selects all special objects - like Flares and Fountains

Lights: Selects all lights in the current space.

Options panel

Deselect First: Deselects any ~~polys~~polygons that are selected before selecting ~~polys~~polygons satisfying the request.

Hide All But Selected: Hide any ~~polys~~polygons that don't satisfy the request.



/// TEK: Duplicated in menu reference (p. 137)
Name: Spiral
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Create spiral polygons.

Documented on page 134.This tool can be used to create single polygons that have a spiral shape. . These are useful for motion paths, or extrusion paths. . They are not generally used directly for rendering. . Selecting the option brings up the Spiral Settings requester.

Segments: This numeric entry box sets the number of segments (or points) in the spiral shaped poly.

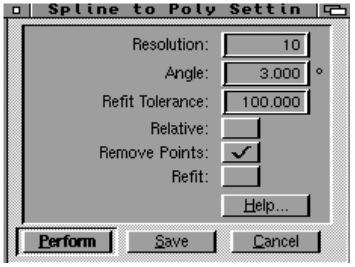
Sweep Angle: the angle in degrees that the poly will wrap when creating the spiral.

Start Radius: sets the beginning radius of the spiral.

End Radius: sets the ending radius of the spiral.

Movement X, Y, and Z: Allows translation along the given axes while the spiral is being generated.

Rotation Axis: (rotary gadgetcycle gadget) choose the axis that the spiral will wrap around.



/// TEK: Duplicated in menu reference (p. 137)
Name: Spline<-->Poly (non-interactive or interactive)
Access: LMB
Secondary: RMB
Keyboard: None

Brief: Convert selected splines to polypolygons and polypolygons to splines.

This tool allows you to convert all selected polygons into splines, and all selected splines into polygons. . Select the tool with the right mouse button to get the tool's settings requester. . If you click the tool with the left mouse button, it performs the operation according to the defaults set in the settings requester. . The gadgets in this window are documented on page 134.:

Resolution: (spline to poly) Here you set the number of points that the program will divide each segment of the spline into when constructing the polygon. . This will be modified by Relative if on, and by Angle if Remove Points is on.

Angle: (spline to poly) Here you specify the angle in degrees that you want as the minimum allowable angle between the sides of the poly created.

Refit Tolerance: If refit is ON, then this number is the maximum distance that a point can be from a spline before it is used to modify the spline being created.

Relative: (spline to poly) If ON, the program will determine a smaller or larger resolution to use, based on the relative length of each spline segment. . The resolution you enter is still used as the base resolution, then a very short segment may get a smaller resolution, and a long segment will get a larger resolution.

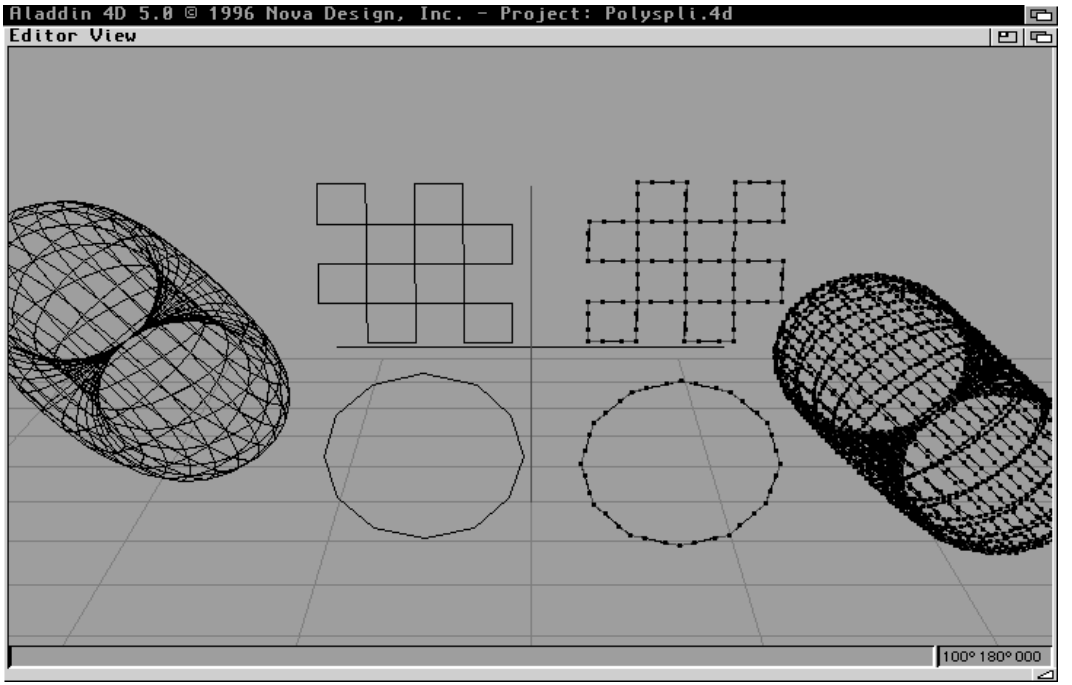
Remove Points: (spline to poly) If ON, the program will remove points if the angle between the adjoining sides of the resulting poly is less than that you specify in the angle gadget.

Refit: (poly to spline) If ON, the program will treat the polygon as though its points are crude samples from a data set, and attempt to rebuild the data. . This results in the spline rounding over any angles.

The conversion is performed on selected polyspolygons AND splines at the same time. . When the conversion is performed, the original spline or poly is not deleted.

When converting polyspolygons to splines, if Refit is OFF, the program makes a spline segment for each side of the polygon. . Control points are in place at one-third intervals for you to edit curvatures if desired. . However if Refit is ON, the program will treat the poly's points as though they are a rough part of a much larger data set and attempt to automatically fit a spline to these points.
Limits: None.

These are some examples. . The polyspolygons are on the left, and the splines are on the right.



screenshots: examples as described /j
HARV NOTE: IT IS BEYOND MY CAPABILITIES TO MAKE THIS ILLUSTRATION.

=====

Hints

In this section are some hints and tips, and some pointers on what to do if something doesn't work.

DEGENERATE ~~Polys~~POLYGONS

A degenerate poly is one that won't render properly~~.~~. There are two types~~.~~. The first is a non-flat poly, and the second has its three sampled points (used to calculate the polygon's "normal") all in a straight line.

A polygon's "Normal" is a 3D mathematical function - similar to a tangent to a circle in 2-D geometry - that forms the basis of rendering engines~~.~~. It is a straight line perpendicular to the plane of the polygon~~.~~. The Normal is essential to equations that calculate the light on the poly, to calculations required for Phong shading, and to find the exact 3-D coordinate for procedural textures.

Both of these types of degenerate ~~polys~~polygons cause trouble because the ~~polys~~polygons' Normals cannot be calculated in the way Aladdin 4D is expecting to do it~~.~~. An easy way to make a degenerate poly is to select one point in a rectangle and move it out of the plane of the other points~~.~~. The Normal is calculated only once for the polygon, and will be correct for the plane described by the first three points in the poly, but wrong for the last point.

You can correct this by flattening out the polygon, or by breaking it down into triangles~~.~~. For a complex polygon (more than four points is "complex"), if there are a lot of points - like the face of a font - the program calculates its Normal using the poly's first point, then a third of the way through, and one that is two-thirds of the way through~~.~~. If these points happen to form a straight line, others are selected until they do not~~.~~. If this can't be done, the Normal is undefined, and the polygon will usually paint black or white, regardless of the light you put on them~~.~~. You ~~can~~ correct this by selecting a new First Point for the polygon.

CIRCULARLY LINKED PATHS

Be careful~~!~~! The render routines in Aladdin 4D no longer tolerate a circularly linked path hierarchy - that's one where the hierarchy's end path points back to the first path (This may be the same path assigned to itself)~~.~~. This can ONLY happen with a drawing that has been saved from an older version of the program~~.~~. Older versions of Aladdin 4D were somewhat forgiving of this error~~.~~. If you load an old drawing and during preview or render, the program puts up a wait pointer that never goes away, you have a circularly linked path - and a crashed computer.

Reboot and reload the drawing~~.~~. Then select the paths, one by one, and select the Path menu item Show Assigned Paths~~.~~. If a circular link exists in the selected path's hierarchy, the program will tell you and attempt to repair it~~.~~. You should then be able to preview and render the drawing~~.~~. Re-save the old drawing after correcting the problem.

Newer versions of Aladdin 4D won't allow creation of a circular path, so you can't put such a problem into a new file - you'll get them only when loading old files.

BITMAP TEXTURE SIZE:

You can use any reasonable size texture. When deciding the resolution to use, keep in mind that any resolution larger than roughly the size the image will be on the screen is probably wasted. Often small textures, such as 16x16-pixel brushes saved from any Amiga paint program, give very pleasing results, using the large pixels to advantage. In general, images of 160x100 up to 1024x768 are acceptable.

BITMAP TEXTURE ERRORS:

There can be several types of errors in bitmaps which are to be applied as textures. The creating program, not Aladdin 4D, causes these. They are:

LEFT FRINGE:

Caused by some types of digitizers, especially flatbed scanners. Either fix the problem in a paint program, or use the index percentages to ignore the left border of the image.

DIFFERENT BRUSHES:

If the brush shows more information when you use it as a texture than the one you cut out of a paint program, the paint program is cutting a larger brush than the one it shows you. Usually, this goes up to the next larger number divisible by 16. DeluxePaint does not do this, but DigiPaint does. You can either correct the brush in another paint program or use Aladdin 4D's Percentage index settings to show only the part you intended.

DARK LINE AT TOP:

The image came from a program that uses information at the top (and sometimes at the left) of the image to enhance its display qualities, or color. DCTV images are a good example of these. DO NOT correct these by cropping them in a paint program - they are probably used by Aladdin 4D for the same purpose. Instead, use the percentage index settings to move the areas out of display in the texture.

ONLY ONE LEVEL OF DEFORM SHOWS:

You haven't given the other levels any time. See the tutorial on animation.

ODD TEXTURE COLORS ON GOURAUD OBJECTS:

Gouraud shaded and faceted objects use the change in color of the base poly to determine the amount of light hitting the poly. Use a base color for the ~~polys~~polygons of at least 100, 100, 100 and the color will be true. The higher the base color numbers, the more accurate the colors.

TRIANGLES AT TOP OF LATHED OBJECT SHADE ODDLY:

You have created a closed Lathed object (like a sphere) and forgotten to turn Last Segment OFF. There is a stem of straight line ~~polys~~polygons in the center of the object. If you delete these, or turn Last Segment OFF, the problem will be solved.

DARK/LIGHT SPOTS, OR HOLES IN OBJECTS DURING RENDERING:

You have created degenerate ~~polys~~polygons (no longer flat). Solve the problem by turning them into triangles.

GENLOCK TEXTURE TYPE DOESN'T WORK:

The Attribute List for the polygons must have a transparency of at least 1 for the Genlock option to work.

ALTERNATING BANDS OR RANDOM AREAS OF COLOR ON EACH SCANLINE:

You have two polygons occupying the same space. This is not permitted. Delete or move one of them.

ANIMATION PLAYBACK SPEED IS SLOW:

After you have saved the animation and load it into your player program, you may find that it won't play back as fast as you like. This is often caused by the number of color changes between frames. If you fade out a background, or pan a camera, you are creating a very large number of changes from one frame to the next. The player must decompress these changes for each frame, which can take longer than the time you want between frames. Try to avoid very large areas of color change when using OpCode 5 (standard Amiga) animations.

A slow running animation will play faster on a faster Amiga, because the decompression of frames happens quicker. If your Amiga is not already powered by one of the heavy-duty microprocessors, like 68040 or 68060, you can substantially improve animation playback speed by upgrading to a faster machine. The Amiga 4000 series computers is capable of much faster animation, partly due to an improved operating system (AmigaDOS 3.x).

For single-frame animations, these limits do not apply, so if you must have the fades and pans that cause this problem, single-framing is the solution.

DCTV TROUBLE:

Aladdin 4D doesn't run without dctv.library, and it can't create pictures for DCTV unless dctv.library exists in your "LIBS:" directory. This is installed with DCTV's software, as well as by some image processing packages that read and write to DCTV's file formats.

Appendix A: Software License

This product; manual and software, are Copyright 1996 by Nova Design, Inc., All Rights Reserved.

ATTENTION: PLEASE READ THIS DOCUMENT CAREFULLY. THE INDIVIDUAL OR ENTITY OPENING THIS PACKAGE (THE "END USER") AGREES TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU OPEN THIS PACKAGE AND DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE SOFTWARE AND PROMPTLY RETURN THE PACKAGE UNUSED, AND THE LICENSE PRICE WILL BE REFUNDED.

The enclosed computer program(s) and the accompanying documentation are provided to the End-User by Nova Design, Inc., a Virginia Corporation
("Licensor") for use only under the following terms. Licensor reserves any right not expressly granted to the End-user. The End-User owns the disk on which the Software is recorded, but Licensor retains ownership of all copies of the Software itself. The End-User assumes sole responsibility for the installation, use and results obtained from use of the Software.

1. License.

End-User is granted a limited, non-exclusive license to do only the following:

A. Install and maintain the Software on one computer at any time for
use only in the End-User's own business.

B. Make one copy in machine-readable form solely for backup or archival
purposes for the computer which the Software is installed. The Software is protected by copyright law. As an express condition of this License, the End-User must reproduce on the copy Licensor's copyright notice and any other proprietary legends on the original copy supplied by Licensor.

C. Transfer the Software and all rights under this License to another party together with a copy of this License and all written materials accompanying the Software, provided (i) the End-User gives Licensor written notice of the transfer (including in such notice the identity of the transferee), and (ii) the other party reads and agrees to accept the terms and conditions of this License.

2. Restrictions.
The End-User may NOT sublicense, assign, or distribute copies of the Software to others. The Software contains trade secrets. The End-User
may NOT decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human readable form. THE END-USER MAY NOT MODIFY,
ADAPT, TRANSLATE, RENT, LEASE, LOAN, RESELL FOR PROFIT, DISTRIBUTE, OR OTHERWISE ASSIGN OR TRANSFER THE SOFTWARE, OR CREATE DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF, EXCEPT AS EXPRESSLY PROVIDED IN SECTION 1.C., ABOVE.

3. Protection and Security.
The End-User agrees to use its best efforts and to take all reasonable steps to safeguard the Software to ensure that no unauthorized person shall have access thereto and that no unauthorized copy, publication, disclosure or distribution in whole or in part, in any form, shall be made. The End-User acknowledges that the Software contains valuable confidential information and trade secrets and that unauthorized use and/or copying are harmful to Licensor.

4. Termination.
This License is effective until terminated. This License will terminate immediately without notice from Licensor if the End User fails to comply
with any of its provisions. Upon termination the End User must destroy the Software and all copies thereof, and the End-User may terminate this License at any time by doing so.

5. Limited Warranty.
Licensor warrants that, for thirty (30) days from the date of initial purchase by the original End User, the Software shall operate substantially in accordance with the published functional specifications current at the time of shipment. If, during the warranty period, a defect appears, End User shall return the Software to Licensor and Licensor's only obligation shall be, at Licensor's election, to replace the defective Software or refund the purchase price. The End-User agrees that the foregoing constitutes the End-User's sole and exclusive remedy for breach by Licensor under any warranties made under this Agreement. This warranty does not cover any Software that has been
altered or changed in any way by anyone other than Licensor. Licensor is not responsible for problems associated with or caused by incompatible operating systems or equipment, or for problems in the interaction of the Software with software not furnished by Licensor.

No oral or written information or advice given by Licensor or its dealers, distributors, employees or agents shall in any way extend, modify or add to the foregoing warranty.

THE WARRANTY AND REMEDY PROVIDED ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE END-USER ASSUMES ALL RISK AS TO THE SUITABILITY, QUALITY, AND PERFORMANCE OF THE SOFTWARE. IN NO EVENT WILL LICENSOR, OR ITS DIRECTORS, OFFICERS, EMPLOYEES OR AFFILIATES, BE LIABLE TO THE END-USER FOR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR EXEMPLARY DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF DATA OR BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR ACCOMPANYING WRITTEN MATERIALS, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

LICENSOR'S LIABILITY TO THE END-USER (IF ANY) FOR ACTUAL DIRECT DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION, WILL BE LIMITED TO, AND IN NO EVENT SHALL EXCEED, THE AMOUNT ORIGINALLY PAID TO LICENSOR FOR THE LICENSE OF THE SOFTWARE.

6. Enhancements.
From time to time Licensor may, in its sole discretion, advise the End-User of updates, upgrades, enhancements or improvements to the Software and/or new releases of the Software (collectively, "Enhancements"), and may license the End-User to use such Enhancements upon payment of prices as may be established by Licensor from time to time. All such Enhancements to the Software provided to the End-User shall also be governed by the terms of this License. IN ORDER FOR THE END-USER TO BE ASSURED THAT IT WILL BE ADVISED OF AND LICENSED TO USE ANY ENHANCEMENTS TO THE SOFTWARE, THE END-USER MUST COMPLETE, SIGN AND RETURN TO LICENSOR THE ATTACHED END-USER REGISTRATION CARD.

7. General.
This License will be governed by and construed in accordance with the laws of the Virginia, and shall inure to the benefit of Licensor and End-User and their successors, assigns and legal representatives. If any provision of this License is held by a court of competent jurisdiction to be invalid or unenforceable to any extent under applicable law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this License will remain in full force and effect. Any notices or other communications to be sent to Licensor must be mailed first class, postage prepaid, to the following address: Nova Design, Inc., 1910 Byrd Ave, Suite 204, Richmond, VA 23230 USA

This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof, and all prior proposals, agreements, representations, statements and undertakings are hereby expressly canceled and superseded. This Agreement may not be changed or amended except by a written instrument executed by a duly authorized officer of Licensor.

8. Acknowledgment.
BY OPENING THIS PACKAGE, THE END-USER ACKNOWLEDGES THAT IT HAS READ THIS LICENSE, UNDERSTANDS IT, AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS.

Should you have any questions concerning this License, contact Licensor at the address set forth above.

Problem Reporting Form

When mailing or faxing Nova Design's support team, please take the time to fill out the following form as completely as possible. We suggest you copy this form from the book rather than tearing it out!

Name: _____

Address: _____

City, State, Zip: _____

Country, Postal Code: _____

Telephone Number(s): _____

Product: _____

Serial Number: _____
(on diskette label)

Version: _____
(in about box)

Describe the problem you are experiencing. Go through it step by step so that our support team can recreate the problem exactly. Provide all relevant information, eg: image size, version of AmigaDOS and CyberGraphX(if applicable), color depth of screen, etc.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

~~Appendix B: 24-bit boards~~

~~@ #Firecracker Clear~~

~~@ Menu Item: Firecracker Clear~~

~~@ Keyboard: None~~

~~@ Purpose: Clear old images from the FireCracker memory~~

~~@~~

~~@ Brief: This menu item allows you to clear the FireCracker's memory~~

~~@ to the current color in register 0,~~

~~@ which you can set in the palette. This is not done automatically, because~~

~~@ you may want to save an image that was in the Firecracker's memory from a~~

~~@ previous rendering session. (Just open the render screen~~

~~@ and IMMEDIATELY abort the redraw with <Esc>. Then use~~

~~@ the Save IFF menu item. The portion of the FireCracker's memory that~~

~~@ corresponds to the render screen resolution will be saved in IFF 24 format.)~~

~~@~~

~~@ The first time you render to the FireCracker each session, you~~

@ will need to do a FireCracker clear if you are in a low-res screen size.

@ Once you have done this, you will not have to do it again until you turn

@ your computer off - or when you want a clear area.

@

@ Limits: None

@

@

@ #Resolver Control

@ Menu Item: Resolver Control

@ Keyboard: None

@ Purpose: Specify resolver parameters

@

@ Brief: This menu item allows you to control the Resolver board, if one is present.

@ To access the Resolver board, you must change the tooltype of Aladdin 4D's icon

@ to enable Resolver support before you start the program.

@ Then when you select this menu item, you'll get the Resolver Control window.

@ Screen size settings may be edited in the width and height gadgets. The board's

@ current mode may be cycled by clicking the Next gadget. These are

@ the modes you have preconfigured for the board with Resolver's setup program.

@

@ Of particular interest here is palette control. If Greyscale is

@ ON, the image will be rendered as a 256 greyscale image, which is very fast,

@ and looks quite photographic in higher resolution modes.

@ If you have Greyscale OFF, the program must find the best 256 colors out

@ of 16,777,216 colors the program works with internally. Since Aladdin 4D

@ is a single pass renderer, these colors must be selected on the fly,

@ scanline by scanline. The program only knows the colors in the current

@ scanline and the colors selected so far, so it allows you to specify the

@ color distance that a new color must be greater than before it

@ is assigned a register color of its own. Edit the Color

@ Distance gadget, and then redraw. Colors are placed at the bottom of

@ the screen as they are selected.

@ You can also set the Color Distance to a large number,

@ limiting the palette to a smaller range.

@

@ To save the image, use the Save IFF menu item. The image will

@ be saved directly from the Resolver's memory as a 256-color IFF.

@

@ Limits: None

////// WHERE IS THIS ??????? 0503 /j

@ #Save Professional Draw Clip

@ Menu Item: Save Professional Draw Clip

@ Keyboard: None

@ Purpose: Save the current view as a Professional Draw clip

@

@ Brief: This menu item allows you to save the current drawing in the view you

@ have in the render screen as a Professional Draw Clip. This allows

@ you to not only use your work as image files, but to also output it as a

@ structured drawing for use in published materials.

@

@ Professional Draw

@ clip files cannot use the shading or texture information from the

@ polygons, so the clip files are saved as either wireframe or full color drawings

@ in the base

@ polygon color. Light will be accounted for, but all drawings will appear

@ faceted in the clip. In spite of these limitations, the results are very

@ impressive - and an excellent way to obtain accurate 3D-perspective for an

@ illustration. For special effect, you can output the clip as wireframe and

@ overlay it onto a 24-bit image.

@

@ When you select this menu item, it opens the Professional Draw Clip Save window.

@ The gadgets in this window are:

@

@ VERT SORT: Professional Draw's polygons are visible in

~~@ a simple back to front order. The main difficulty in saving Professional Draw clips is~~

~~@ ordering the polygons. Professional Draw cannot handle intersecting polys~~

~~@ easily. If the Vert Sort~~

~~@ option is ON, polygons at the center of the screen vertically the same~~

~~@ distance from the viewer as polys at the edge of the screen will be~~

~~@ considered to be in front of the polys at the edge of the screen vertically.~~

~~@~~

~~@ HORIZ SORT: As above, but horizontal position is used for compare.~~

~~@~~

~~@ BLACK OUTLINES: If ON, all polygons in the drawing will be saved~~

~~@ with black outlines. If off, the color and light intensity of the~~

~~@ polygon will be used for both the polygon and the outlines.~~

~~@~~

~~@ BLACK: You may not want black in your clip. If not, turn this OFF.~~

~~@~~

~~@ WHITE: If you don't want white in your clip, turn this OFF.~~

@

@ Professional Draw 2.0: If this is ON, the clip is saved with a palette of the full

@ 16-million-plus colors used in Aladdin 4D. If this is OFF, the clip is saved

@ with the limited palettes required by versions of Professional Draw prior to

@ 2.0. If you're producing clips for black outline printing

@ only, this may be left OFF for smaller clips.

@

@ COLOR: This option is only available if the Professional Draw 2.0 switch is on. It

@ allows the clip to be saved in full color.

@

@ POLY REM SENS: If you intend the clip to be solid, not wireframe, it is

@ not necessary to save polygons that are completely covered by others.

@ This value tells the program how many pixels a polygon may occupy

@ on the screen before it is considered to be visible. If this value is 0, a

@ polygon will be in the clip if it occupies one or more pixels on the screen. If

@ this value is 2, a polygon will be in the clip if it has "more than 2"

@ pixels on the screen. Removing polygons from the clip in this way

@ makes for smaller clip files and faster handling in Professional Draw.

@

@ CLIP NAME: This is the name you wish Professional Draw to recognize for the clip.

@

@ SAVE TO: This is the path and filename you wish to save the clip to.

@

@ ABORT: Close the window without saving the clip

@

@ BEGIN: Close the window and proceed to save the clip.

@

@ Using Professional Draw clips for illustration of your story boards makes

@ good-looking, structured story boards that you can re-size at will without

@ either the overhead or the potential problems of handling bitmaps. The

@ format is recognized by Professional Page and PageStream

~~@~~

~~@ ///FLAG: TRADEMARK /j~~

~~@~~

~~@ page layout programs, and can be converted to other formats by other Amiga structured~~

~~@ drawing programs, such as ProVector.~~

~~@~~

~~@ You can redraw any frame in Aladdin 4D and save it as a Professional Draw clip.~~

~~@ When you load it into your page layout program, however, the clip will be~~

~~@ scaled to fit the box you load it into. You can correct this at will,~~

~~@ without worrying about the "jaggies" you'd be likely to get with bitmaps.~~

~~@~~

~~@ For story boarding, load the clip into Professional Draw first, draw a box~~

~~@ around the area of the clip you want in your illustration,~~

~~@ and resave the clip. Then in your page layout program, use this box as the~~

~~@ outside frame, insuring the motion represented in the story board frames~~

~~@ is the same scale, and positions are relative to the box.~~

@ Polygons saved in the clip are included in their entirety, not clipped to

@ the screen, and will appear unless they are completely off the screen.

@

@ Limits: As discussed above, Professional Draw cannot handle textures

@ and shading information so they are ignored. Some clips may have ordering

@ errors and will have to be manually adjusted in Professional Draw unless they

@ are wireframe black only. Splines and such are not saved, since these are

@ non-rendering objects.

@ #Hide Unshown

@ Menu Item: Hide Unshown

@ Keyboard: None

@ Purpose: Return to Editor hiding unshown polys

@

@ Brief: This menu item will return you to the Editor, but when you get there,

@ any polygons that did not appear in the render will be hidden. If a polygon

@ has one or more pixels occupied in the render screen, it will be visible in

@ the Editor. If it had no pixels occupied, it will be hidden.

@ This is of particular use in saving Professional Draw clips, because you can

@ render the

@ screen, then use this menu item to see exactly which polygons will be in

@ the clip. When back in the Editor, you can hide or show polys to be even

@ more accurate in the inclusion of polys in the clip.

@

@ Limits: None

@

@

@

=====

@ #Palette

@ Menu Item: Palette

@ Keyboard: <P> /// This does not work 0503 /j

@ Purpose: Alter current palette

@

@ Brief: This item gives access to the current palette. When you

@ select it, the Palette window opens. The palette you see will depend on the

@ current display mode you are using. A 64-color palette is shown (Ham-8 mode).

@ If you're in a standard Amiga mode, it is the number of register colors

@ supported for that display mode. For example, 2 for a 2-color screen, 4 for

@ a 4-color screen, 8 for an 8-color screen, 16 for a 16-color screen, 32 for

@ a 32-color screen, and 16 for a HAM screen.

@

@ If you are in a DCTV screen, you can have either 8 or 16 colors (3 or 4

@ bitplanes).

@

@ In 24-bit card mode, or Resolver mode, the program defaults to

@ a 320x200 4-color screen - a formality,

@ to support menu selection and give you access to a palette so you can change

@ color 0, which is used as the background color for the true displays which

@ come from these boards. Since color 0 is transparent for these boards, swap

@ color 0 with color 1, edit color 1, then reswap color 0 with color 1. This

@ way you can see the color you are requesting.

@

@ The gadgets in the window are:

@

@ R G B SLIDERS: Adjust the red, green and blue components of the

@ current register

@

@ R G B NUMBERS: Manually enter a number for each color component.

@ Range is 0 to 255.

@

@ DEFAULT: Change registers to the default palette

@

@ UNDO: Revert to the condition of the palette before last change

@

@ SPREAD: A limited spread. When selected, spreads the palette from color

@ 1 to the maximum register in use. Designed to give you value oriented

@ palettes for the non-Ham Amiga modes. Aladdin 4D treats any standard Amiga

@ modes that are not Ham as value-oriented displays. Darker polygons are given

@ the colors in the lower registers and brighter polygons are given the colors

@ in the higher registers. This allows you to do color specified wireframes by

@ carefully controlling the reflectivity of the polygons.

@

@ COPY: Copy currently selected color to a destination

@

@ EXC: Exchange currently selected color with a destination color

@

@ COLOR TILES: This is where you select the color you want to change.

@ If you are rendering in Ham modes, you can select colors

@ straight from the Render screen. If you select a color from the render

@ screen, it will replace the active color. This may

~~@ cause the render screen to change wildly on any scanline where the register~~

~~@ color is in use. Normally for the best palette, you would render the view~~

~~@ using an all black palette.~~

~~@ This prevents the program from picking any register colors for the image~~

~~@ except color 1. Then you would open the palette and leave color 1 alone,~~

~~@ but pick color 2 and replace it with a commonly occurring color from the~~

~~@ screen, then do the same for the rest of the colors. Again, except for color~~

~~@ 1. Then turn on the palette match and render. You will see one of the best~~

~~@ possible Ham screens, with absolutely NO color bleeding.~~

~~@~~

~~@ ABORT: Close the palette with no changes~~

~~@~~

~~@ USE: Close the palette with the changes you have made.~~

~~@~~

~~@ Limits: None (but see the note about color 0 with 24-bit boards).~~

~~@~~

~~@~~

~~@~~

~~@ #Image Control ======= REMOVED ??? 0503 /j~~

~~@ Menu Item: Image Control~~

~~@ Keyboard: <I>~~

~~@ Purpose: Specify general aspects of the image~~

~~@~~

~~@ Brief: This menu item allows you to specify some general aspects of the~~

~~@ image, such as brightness, contrast, etc. This item opens~~

~~@ the Image Control window. The gadgets in this window are:~~

~~@~~

~~@ Color Level: A positive number will increase the amount and purity of~~

~~@ color in the rendering. A negative number will decrease it. With~~

~~@ the slider all the way to the left, the drawing is rendered as a greyscale~~

~~@ - quite useful if it is to be printed as a greyscale.~~

~~@~~

@ Brightness Level: A positive number increases brightness; negative

@ numbers decrease brightness.

@

@ Contrast Level: A positive number increases contrast in the image;

@ negative numbers decrease contrast.

@

@ Tint Level: If desired, you can specify that an amount of the color you

@ have picked in the palette as Color 0 (the background palette color) be

@ added to the image as a final "wash". This can make subtle differences in

@ the image, helping to tie together two such disparate objects as a rendered

@ polygon object and a digitized background.

@

@ Reset: Resets color, brightness and contrast levels to 0.

@

@ Limits (gadget): These values allow you to specify lowest and highest values

@ for each of the red, green and blue color components during the render. This

~~@ may be used to considerable advantage for hardware that allows "keying"~~

~~@ effects based on luma. It can also help you clamp colors to conform to NTSC~~

~~@ (or PAL) broadcast limits. If you use a value of, for example, 10 in the Low Limit~~

~~@ and 250 in the High Limit, as the image is rendered, the colors that should~~

~~@ be in the bitmap are rescaled to the restricted color range. Of course,~~

~~@ instead of rendering the image in the full 16,777,216 colors, the image will be~~

~~@ rendered in fewer colors as determined by the range you permit.~~

~~@ Normally, the Low Limit is 0, and the High Limit is 255, for 256 possible~~

~~@ shades of red, green and blue - 256x256x256 = 16,777,216 colors. If you~~

~~@ set Low=10 and High=250, you allow 241 shades of red, green~~

~~@ and blue, or 241x241x241 = 13,997,521 colors in the final image.~~

~~@ With such a settingm the colors present in the final image will never be~~

~~@ less than RGB Values of 10, 10, 10 and never greater than 250, 250, 250.~~

@ Such a restricted palette - actually, reduced even more - would be much closer

@ to conformity with NTSC broadcast limits. NTSC limits are actually not so easily

@ conformed to as this, but such limits applied at the rendering stage

@ would help out the process greatly.

@

@ Negative: This gadget when selected will cause a complete inversion in

@ your colors, creating the negative of the image.

@ #Ham Dither Lev

@ Menu Item: Ham Dither Lev

@ Keyboard: None

@ Purpose: Specify the amount of dither used in HAM screens

@

@ Brief: This menu item allows you to specify the amount of dither used in the

@ Amiga's HAM mode. The dither type used is a noise dither.

@ When you select the menu item you open the Ham Dither window.

~~@~~

~~@ The usable range of the dither level is 0.0 (no dither) to 1.0 (nothing but~~

~~@ noise.~~

~~@~~

~~@ Limits: None~~